AFRL-RI-RS-TR-2012-085

ADVANCED SUBSPACE TECHNIQUES FOR MODELING CHANNEL AND SESSION VARIABILITY IN A SPEAKER RECOGNITION SYSTEM

CLARKSON UNIVERSITY

MARCH 2012

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2012-085 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

DARREN M. HADDAD
Work Unit Manager

/s/

WARREN H. DEBANY, JR., Technical Advisor
Information Exploitation and Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
March 2012	Final Technical Report	September 2010 – September 2011

4. TITLE AND SUBTITLE
Advanced Subspace Techniques for Modeling Channel and Session Variability in a Speaker Recognition System

5a. CONTRACT NUMBER N/A

5b. GRANT NUMBER FA8750-10-1-0231

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jeremiah Remus

5d. PROJECT NUMBER ASID

5e. TASK NUMBER BA

5f. WORK UNIT NUMBER 01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Clarkson University
8 Clarkson Avenue
Potsdam, NY 13699

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/Information Directorate
Rome Research Site/RIGC
525 Brooks Road
Rome NY 13441

10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TR-2012-085

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The robustness of any speaker recognition system is dependent on its capability for managing the variability in the recording environment. A better ability to quantify that variation may lead to the development of improved methods for reducing the non-speaker influences on performance. In this study, subspace decomposition in combination with three pattern classification techniques was investigated to assess its appropriateness for performing speaker recognition on the MultiRoom8 corpus, a data set with several room and microphone conditions. A partial least squares decomposition of the GMM supervector in combination with a nearest neighbor classifier was consistently a top-performer on the 100 experimental setups consider in this study, which may suggest an approach for mitigating the effects of room and microphone variability in a speaker recognition system through projections to a lower-dimensional feature space.

15. SUBJECT TERMS
Speaker Identification, room variability, channel variability, pattern classifications

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	53	DARREN M. HADDAD

19b. TELEPHONE NUMBER (*Include area code*) N/A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 SUMMARY

Speaker recognition methods have long been capable of verifying a speaker's identity when two speech recordings come from similar or identical environments and channels. A significant focus in recent years has been the development of methods for extending the performance of speaker recognition systems to scenarios where greater variation between the enrollment and test data may exist. The most common approach for managing these sources of non-speaker variation is adoption of a model that assumes the captured recording is a superposition of two elements: the speaker-specific features useful for speaker recognition and an additive non-speaker term. Several investigators have proposed linear subspace modeling techniques that can be used to estimate and factor out the non-speaker component in recorded audio.

This technical report describes a research effort that investigated an approach to linear subspace modeling applied to the sponsor-provided MultiRoom8 corpus. This data set consists of 51 speakers recorded in ten different conditions, with each condition defined by a unique combination of room and microphone. Four rooms (conference room, small room, medium room, and large room) and five microphone configurations using an omnidirectional and directional microphone at different distances provided diverse sources of environmental variability. Several variations on the standard speaker recognition approaches were considered in this study. Baselines levels of performance were first established using the standard GMM-UBM and GMM supervector as an input feature for three different pattern recognition methods. One of the pattern recognition methods, the linear-kernel support vector machine using the GMM supervector as input features, has performed very strongly in recent speaker recognition evaluations. A primary avenue of investigation in this study was the use of partial least squares (PLS) to decompose the GMM supervector, resulting in a significantly lower-dimensional representation in a subspace that would be better-suited for discriminating individual speakers. The three pattern recognition techniques considered in this study were applied to both the high-dimensional GMM supervector and much lower dimensionality PLS projected subspace for comparison of the discriminability in the two feature spaces.

The results of this study indicated that the partial least squares (PLS) subspace consistently provided a better feature set for discrimination between speakers. These results were generated for 100 different experiment configurations created by using each of the ten conditions in the MultiRoom8 data set separately as training and testing data. In the PLS subspace, the nearest neighbor classifier with a correlation-based distance metric provided the best performance, with lower equal-error rates than the support vector machine and Random Forest classifier applied to the same features. The PLS – Nearest Neighbor classifier also outperformed the GMM supervector SVM; thus, it was the best performing method for discriminating between the MultiRoom8 speakers that was considered in this study.

The results of this study provide further evidence to support the validity of partial least squares decomposition for mitigating certain sources of variability in speaker recognition tasks. Previous

studies have also shown that partial least squares decomposition provides a lower-dimensional subspace that is appropriate for discriminating between speakers [1]. The outcomes of this research effort encourage further consideration of supervised subspace decomposition techniques (e.g. partial least squares) to address scenarios where speaker recognition must be performed in the presence of significant room and microphone variability.

2 INTRODUCTION

There has been substantial interest in the effect of session variability on speaker recognition systems. Session variability can be attributed to a number of possible sources: variation in the speaker's voice due to illness, aging, or stress condition; recording environment (*i.e.* background noise level); and changes in the recording channel (*i.e.* cellular phone versus landline handset). One approach for handling session-to-session variability that has received significant attention is a decomposition of the feature supervector into two components.

$$M(s) = m(s) + Ab(s) \tag{1}$$

In such a framework, the recording from a speaker is represented as a feature supervector $M(s)$ that is considered to be a superposition of a speaker model $m(s)$, which is independent of the session conditions, and the term $Ab(s)$, which accounts for the session variability. There have been a number of approaches developed in recent years that utilize this basic framework, differentiated by the various assumptions that are made while estimating the model parameters. Eigen-voice methods [2] assume that the feature supervector $M(s)$ is constrained to a linear low-dimensional "speaker space". This assumption significantly reduces the computational complexity and reduces the time required for speaker adaptation. The eigen-voice method was combined with extended maximum *a posteriori* (EMAP) estimation to produce the eigen-channel MAP method [3]. In this method, the EMAP estimation is used to include the correlations between Gaussian components in the model, adding computational complexity but also greater modeling power. The resulting decomposition is similar to the feature mapping of Reynolds [4]; however, Reynolds assumes a discrete set of channel conditions whereas all of the eigen-methods allow for a continuous representation of channel effect. Alternatively, rather than performing speaker adaptation, Vogt et al. [5] developed a more direct model of session variability that removed the need for discrete classification and labeling of channel conditions. The most significant changes in Vogt et al. are reflected in the assumptions regarding subspace dimensionality and the manner of using the training data. This technique provides the basis for the GMM latent factor analysis (LFA) used in the MIT Lincoln Laboratory 2008 Speaker Recognition System [6].

Altogether, previous work in speaker recognition provides substantial empirical evidence that decomposition of the feature vector into speaker and non-speaker components is an appropriate approach to mitigating the problems caused by session variability. There has been a significant amount of work towards dimensionality reduction, using techniques such as Principal

Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Latent Symantec Analysis (LSA). The significance of this study is to assess the effectiveness of dimensionality reduction techniques on performance for a speaker recognition system in the presence of environment-based variability. This study will also examine the use of partial least squares (PLS), which has only recently seen use in the speaker recognition community [1], as well as a proposed method for nonlinear dimensionality reduction. The dimensionality reduction techniques will be used in conjunction with pattern classification method such as the support vector machine (SVM), nearest neighbor, and Random Forest classifier. The state of the art automatic speaker recognition system performs relatively well on channel mismatch, but other environmental factors including room variability may still pose a significant challenge.

This study utilized the MultiRoom8 data set, made available for this project by the sponsor. The MultiRoom8 data set consists of multi-session audio recordings with collection conditions designed to include a number of distinct environmental scenarios (e.g. noise and room acoustics). A total of 424 audio recordings were used in this study, each approximately three minutes in duration (the data collection procedure was based on an interview scenario). As part of the experiment setup process, each audio recording was divided into two segments of equal length to allow training and testing within a condition (since only one recording was available per speaker per condition). The environments in MultiRoom8 utilized in this study include three distinct rooms of various sizes: small (206 ft^2, 19 m^2), medium (430 ft^2, 40 m^2), and large (2013 ft^2, 187 m^2). There were five microphone/recording setups available, although not all were available in each environment. In the small, medium, and large rooms, directional and omni-directional microphones were located at a range of distances from the speaker. From the available data, a set of ten conditions were selected for analysis of the variability introduced by different room and microphone types. The audio files used in this study were collected from a group of 51 speakers with 35 speakers common to all ten of the conditions. Table 1 lists the number of speakers present for each room/microphone combination. There are four rooms, distinguished by size, and two types of microphones (directional and omnidirectional) at different distances (3 ft, 5 ft, close, mid-distance, and far). In Condition E, the directional microphone at a distance of 5 feet is pointed away from the speaker.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

The research effort for this project can be divided into three experiment groups: 1) a baseline study using the GMM-UBM classifier and evaluation of the GMM-UBM parameter settings appropriate for the MultiRoom data set, 2) evaluation of speaker recognition techniques that utilize the GMM supervector as input features, and 3) evaluation of the effect of supervector decomposition techniques on speaker recognition system performance. Thus, the three experiment groups are organized by increasing sophistication, from baseline GMM-UBM techniques that are well-established in the speaker recognition community and progressing to more recently proposed methods for supervector decomposition.

Table 1. Number of speakers representing each room/microphone combination.

Condition	Room	Microphone	Number of speakers
A	Oasis	Omni @ close	51
B	Small	Dir @ 3ft	39
C	Medium	Dir @ 3ft	44
D	Large	Dir @ 3ft	39
E	Small	Dir @ 5ft	42
F	Medium	Omni @ close	44
G	Small	Omni @ mid-dist	39
H	Medium	Omni @ mid-dist	44
I	Small	Omni @ far	43
J	Large	Omni @ far	39

The SPro and ALIZE/LIA [7] open-source toolboxes provided the primary framework for implementation of the speech feature processing and speaker recognition algorithms. These tools provided the code basis for turning the .wav audio files in the MultiRoom data set into GMM-UBM models, which were a necessary component for all of the studies performed in this research effort.

3.1 Baseline GMM-UBM processing

For all of the experiments performed in this research effort, the feature sets that were used were in some way derivatives of the GMM-UBM model. Thus, the baseline GMM-UBM plays an important role in all of the proposed work. Figure 1 illustrates the processing of computing the GMM-UBM for any .wav recording (to be used for either enrollment or verification).

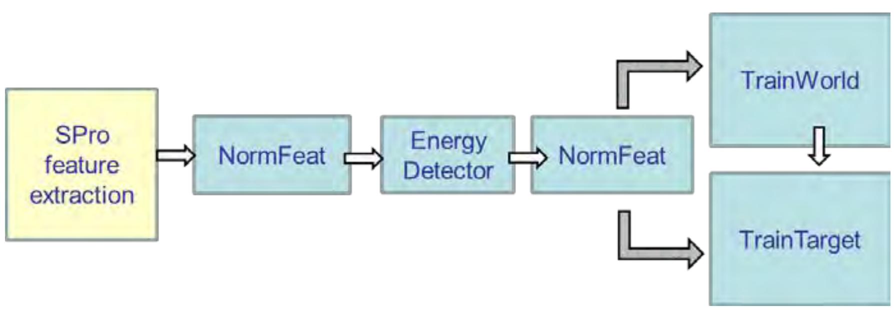

Figure 1. General processing flow for generation of the GMM-UBM model

Each of the functions within the GMM-UBM generation process contains parameters critical to the final model. The values for the parameters were determined based on the speaker recognition literature and through sensitivity analysis conducted using the MultiRoom data. In this study, the GMM-UBM setup used a 32-element feature vector constructed from 16 MFCC coefficients and the first order derivatives. Cepstral coefficients were extracted from 20 millisecond frames with 50% overlap. Once the cepstrum coefficients were extracted, the matrix of data for each

recording was normalized and silent frames were removed using the ALIZE/LIA "NormFeat" and "EnergyDetector" functions. The "NormFeat" function standardizes each column of cepstral coefficients to be zero-mean and unit-variance, and it is run both before and after the "EnergyDetector" silent-frame removal stage. The "EnergyDetector" function determines clusters of frames using a Gaussian mixture model and the highest energy frames are selected based on the weighting parameters of the Gaussian mixture model. After normalization and silence removal, the MFCC coefficients were used to either estimate a universal background model (UBM) or adapt a Gaussian mixture models (GMM), depending on whether the input .wav file is from the development data set or if it is to be used for training/testing. The UBM was generated from 100 separate speaker files containing more than five hours of speech. A 750-component diagonal covariance GMM was fitted to each of the MFCC representations.

A sensitivity analysis was conducted on the GMM-UBM Model to determine the relationship between the many variables in the model and the model output. The GMM-UBM parameter sensitivity study focused on parameters in the energy detector, UBM training, and GMM adaptation stages. The first effort in the sensitivity analysis focused on the potentially influential parameters within the "EnergyDetector" function, which are listed in Table 2 along with the range of values to be considered:

Table 2. Parameters within the "EnergyDetector" function of the GMM-UBM that were examined in the sensitivity study.

Parameter	Range of values
minLLK	[-500, -50]
maxLLK	[50, 500]
nbTrainIt	[5, 20]
varianceFloor	[0.0001, 1000]
varianceCeiling	[1, 10]
mixtureDistribCount	[2, 20]
baggedFrameProbabilityInit	[0.001, 1]
thresholdMode	{weight,meanStd}
alpha	[0, 1]

There were 69,120 possible combinations of the parameters using the ranges of values provided in Table 2. A subset of 10,000 of the combinations was downselected and five .wav recordings were processed through the first four stages in Figure 1 using each of the 10,000 sets of Energy Detector function parameters. The number of frames extracted from the .wav recording were observed and stored for later analysis to examine the relationship between the extracted cepstral feature set and the Energy Detector parameters.

The second sensitivity study examined the relationship between the performance of the speaker recognition system and 1) the mixtureDistribCount in the "EnergyDetector" function (found to be the primary significant parameter) and 2) the number of mixtures in the GMM-UBM model.

The number of mixtures used in Energy Detector were selected from the set {2,4,6,8,10} and the GMM-UBM model selected its number of mixtures from the set {500, 750, 1000, 1500}. A cross-condition speaker recognition experiment (using all available conditions in the MultiRoom data set) was run for all 20 combinations of parameter values. The equal-error rates (EER) for all conditions were stored and analyzed to investigate the preferred parameter values.

As an alternative to performing verification and recognition using the GMM-UBM models, an approach that has recently grown in popularity is generation of a GMM supervector for use as a feature. The GMM supervector is generated by concatenating the means of each Gaussian component from the GMM for a corresponding .wav file. This produces a vector with a number of elements equal to the product of the number of cepstral coefficents and the number of GMM-UBM components. In this study, there were 32 cepstral coefficients with 750 GMM components resulting in a GMM supervector with length 24,000. The GMM supervectors were the basis for the second set of experiments conducted in this research effort, and they provide a strong feature set that is well supported by the literature to allow the use of pattern recognition techniques for discriminating between individual speakers.

3.2 Pattern Classification Techniques

The GMM supervectors were used as features for input to one of three pattern recognition techniques. The purpose of the pattern recognition techniques is to perform a mapping between the GMM supervectors and the individual speaker labels. All of the pattern recognition techniques considered in this study have characteristics that make them particularly suitable for the speaker recognition task: they can estimate any necessary parameters with a single training vector per speaker, they can be configured to manage mapping to multiple speakers (i.e. perform speaker identification), and they can operate in the high-dimensional feature space without significant concerns about computational complexity or ill-conditioning. The three pattern recognition techniques that were applied to the GMM supervector features were 1) the nearest neighbor classifier, 2) the support vector machine, and 3) the Random Forest classifier.

Table 3 lists some distinguishing characteristics for the classifiers considered in this study. Local classifiers make a classification decision based only on the neighboring training samples, whereas aggregate classifiers rely on parameters that are calculated from all of the available training data. A simple test for whether a classifier uses "local" or "aggregate" training data can be conducted by analyzing whether the classifier's output for some test sample x_{TEST} would be sensitive to the addition of a large amount of new training data at a point in feature space not near x_{TEST}. If the classifier's output is not affected by the addition of new training samples, the classifier makes "local" decisions. The second classifier property specified in the table is parametric versus nonparametric classifiers. Parametric classifiers make use of a model to condense the information in the training data to a finite number of parameters, whereas a nonparametric classifier preserves the entire set of training data for making decisions on test

Table 3. Characteristics of the three pattern classification methods considered in this study.

Classifier	Acronym	Local / Aggregate	Parametric / Nonparametric	Reference
Nearest Neighbor	NN	Local	Nonparametric	[8]
Support Vector Machine	SVM	Aggregate	Parametric	[9, 10]
Random Forest	RF	Local	Parametric	[11]

data. Thus, the storage requirements increase for nonparametric classifiers as more training data is acquired.

The nearest neighbor classifier assigns labels to new feature vectors through a fairly straightforward and intuitive process. The distance is calculated between the unlabeled test sample and all of the available labeled training data. The label of the nearest training sample (i.e nearest neighbor) is assigned to the test sample. This classification rule is supported by theoretical results that relate it to nonparametric modeling of probability distribution functions and the likelihood ratio test [8]. For the present study, the negated value of the correlation coefficient was used as the measure of distance between two GMM supervectors.

The support vector machine (SVM) is a sparse, linear, kernel machine. The kernel mapping function can potentially be used to introduce nonlinearity and transform the data into a higher dimensional space where it may be separable by a hyperplane. The SVM finds a decision boundary with the constraint of maximizing the margin, and identifies a small set of "support vectors" that define the decision boundary (and also determine the value of the margin since they are the closest vectors to the decision boundary). Since the SVM utilizes only a small number of training samples as support vectors, it encourages sparseness, and rather than storing all of the training data the SVM only requires a limited subset of training vectors to discriminate between classes. In this research effort, the LIB-SVM [9] implementation was used with two kernel configurations: a linear kernel when operating on the GMM supervectors as features and a radial-basis function (RBF) kernel with unit variance when operating in a low-dimensional subspace generated by linear projection of the GMM supervectors. The SVM is intrinsically configured for binary classification (where only two classes of data are present). To extend the SVM to the current application where many speakers are present, a set of $(N(N+1))/2$ SVMs was constructed, with each SVM discriminating between a pair of the N total speakers in the data set. The $(N(N+1))/2$ classifiers then vote on the final classification of a test sample.

The Random Forest classifier is an ensemble classifier that votes amongst decision trees generated with each node using randomly-selected features [10]. Each individual decision tree splits the data using a subset of features at each node, and continues splitting until it is overtrained to achieve zero empirical error (i.e. perfect classification of the training data). The

set of decision trees are not highly correlated, however, because of the random selection of a feature subset (e.g. two-dimensional) of the available features for use in each node. Therefore, the individual trees within the Random Forest classifier will be more uncorrelated as the number of available features is increased. Assigning labels to new test samples occurs by having the decision trees in the Random Forest vote, and the effects of overtraining will be mitigated by the fact that each decision tree is overtrained differently (due to the random selection of features used at each node and the low correlation between trees). Pilot studies indicated that speaker recognition system performance using the Random Forest classifier was not highly sensitive to the number of component decision trees, so a forest size of 400 trees (near the middle of the range of values investigated in the pilot study) was chosen for use in this study.

3.3 GMM Supervector Decomposition

The second group of experiments focused on the use of GMM supervectors as features for input to pattern recognition techniques, which have been shown in previous studies to provide sufficient information for use in speaker discrimination [11]. However, the supervectors are extremely high-dimensional, which introduces potential concerns about computational complexity and overfitting during the learning stages in the pattern recognition techniques. More significantly, these supervectors contain non-speaker artifacts introduced by the channel, environment, and session-to-session variability. These factors motivate the use of subspace decomposition techniques to find a lower-dimensional representation of the GMM supervector that represents only the speaker-specific attributes and will be robust to variability introduced by changes in channel, environment, and session. An illustration of the desired effect from the subspace decomposition is shown in Figure 2. In the high-dimensional supervector space, several speakers may be indistinguishable due to non-speaker sources of variability. The ideal subspace decomposition would project the supervectors into a lower-dimensional space where all recordings from a single speaker cluster together, and different speakers are separable. Two techniques were considered as part of this research effort: partial least squares (PLS) decomposition and classification-directed dimensionality reduction (CDDR).

Partial least squares (PLS) was originally developed within the chemometrics community, but has since been applied to a variety of topics including bioinformatics (e.g. [12]) and medical diagnosis (e.g. [13]), and more recently speaker recognition [1]. It is most frequently applied as a regression method to model the relationship between a set of independent variables (i.e. features) X and dependent variables Y.

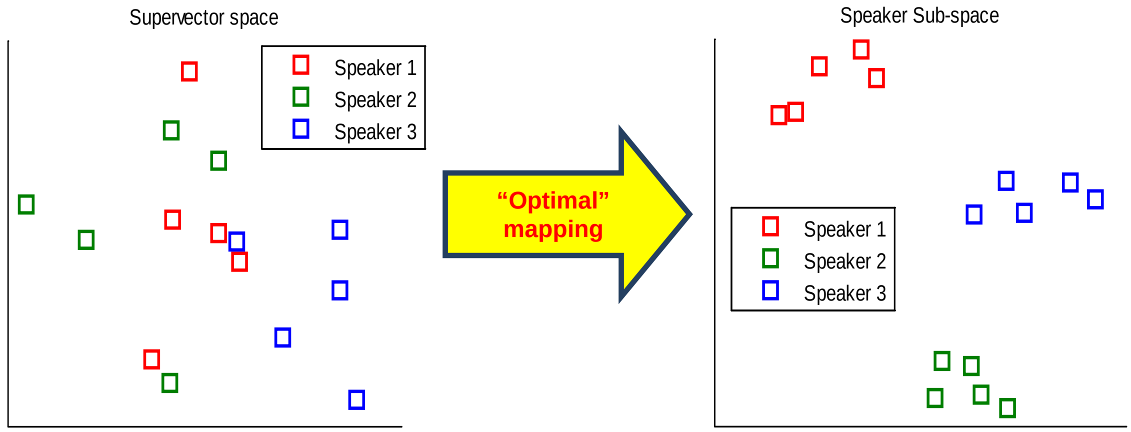

Figure 2. Illustration of ideal subspace decomposition

Partial least squares performs a linear projection to a lower-dimensional subspace, which allows use on high-dimensional data sets without running into the large p, small n problem. One advantage of partial least squares over other linear subspaces projection methods such as principal component analysis is that partial least squares method utilizes the data labels. Therefore, the resulting lower-dimensional subspace is more likely to maintain separability between classes, by using a criterion that seeks linear projections w and q that maximize the covariance between the independent and dependent variables X and Y, respectively, in the lower-dimensional projection space.

$$\max_{\|w\|=1, \|q\|=1} \operatorname{cov}\left(Xw, Yq\right) \tag{2}$$

This contrasts with PCA, which maximizes the variance of the data under the constraint of a unit-norm weight vector, ignoring any available class labels for the training data.

The proposed classification-directed dimensionality reduction (CDDR) generates a matrix of similarities using classification techniques applied to high-dimensional data sets. Existing methods for nonlinear dimensionality reduction techniques (e.g. Isomap [14]) operate on a matrix of distances to neighboring points, and may suggest an approach for finding a lower-dimensional subspace that allows data visualization or may be more conducive for identifying clusters. Unlike traditional manifold learning methods that rely on distances measures for construction of a similarity matrix, CDDR uses classification methods which may be less susceptible to effects of operating in high-dimensional data spaces. Thus, robust operation in high-dimensional data spaces is one of the characteristics that should be considered when selecting a classification method for use in CDDR. Classifiers such as Random Forest [10] and Partial Least Squares Discriminant Analysis (PLSDA) [15] would be appropriate to consider in such conditions.

There are two steps in the classification-directed dimensionality reduction. The first step in the classification-directed dimensionality reduction is generation of an n by m classification table, where n corresponds to the number of in a development data set and m is the number of discrete classes for some physically relevant variable (e.g. speaker ID, gender, and environment). The class label set used in CDDR is selected by the operator, and many relevant label sets can potentially yield good results.

After the classification first stage in CDDR, the result is the classification table, with the (i^{th}, j^{th}) entry corresponding to the probability that observation i belongs to class j. To achieve this in some experiment setups (or with certain classifiers), it may be necessary to have multiple feature vectors from each observation, such that class-membership probabilities can be estimated based on the proportion of features vectors assigned to each class. Once the classification table T has been successfully generated, it can be decomposed using principal component analysis (PCA). Thus, the final CDDR decomposition parameterization consists of the parameters for the classifier in the first stage (i.e. Random Forest) and the PCA loadings in the second stage.

The subspace projection from the high-dimensional supervector features to a lower-dimensional space requires a set of parameters which must be estimated from some data. To provide a more robust experiment result, the GMM supervector decompositions were learned using a separate development data set. Given the available data, there are several possible methods by which the development data set could be constructed: the speakers may be either the same or different from those in the training and test data sets, and the room/microphone combinations may either be the same and/or different from those in the training and test data sets. Table 4 lists the four development data sets that were considered in this study. While the table is constructed with the example of training on Condition A and testing on Condition B, the development data set is adjusted as necessary within the iterations of the cross-condition training and testing as all ten available conditions are eventually used for both training and testing.

Table 4. Composition of the development data sets used to learn the projection coefficients for GMM supervector decomposition.

Development Data Set	Conditions C - J, Subjects 1 to 51	Conditions A and B, Subjects 11 to 51	Conditions A - J, Subjects 11 to 51	Conditions C to J, Subjects 11 to 51
Training Data Set	Condition A, Subjects 1 to 10	Condition A, Subjects 1 to 10	Condition A, Subjects 1 to 10	Condition A, Subjects 1 to 10
Testing Data	Condition B, Subjects 1 to 10	Condition B, Subjects 1 to 10	Condition B, Subjects 1 to 10	Condition B, Subjects 1 to 10
Development Data Set includes:	*Same speakers, different conditions (exclude train/test)*	*Different speakers, train/test conditions*	*Different speakers, all available conditions*	*Diferent speakers, different conditions (exclude train/test)*

3.4 Experiment design and list of experiments

The research effort included several classification experiments to evaluate and assess the performance of different techniques within the speaker recognition framework. There were several parts of the experiment methodology that were common to all of the experiments. Each of the speaker recordings in the sponsor-provided MultiRoom8 data set were divided into two segments by splitting each .wav file at the midpoint in the recording. The set of first-half segments were used for all training and development data set needs, and the set of second-half segments were always reserved for testing and evaluation. Since the microphones for each room (small, medium, and large) were recorded simultaneously, this division of each file into two segments will prevent comparison of recordings on matching text. Potentially more significant is that any anomalous events (i.e. room ventilation switching on, speaker clearing their throat) should not occur in both training and testing files with the same regularity.

For the subspace decomposition methods, the first ten speakers (organized by speaker ID) were used for training and testing. This preserved the higher-numbered speakers for the development set when the development set was to contain a different set of speakers than used for training and testing. Another common element of the experiment setup was the use of cross-condition testing. Each of the ten room/microphone conditions listed in Table 1 were used both for training and testing against all of the other conditions. Thus, results in the form of 100 detection-error trade-off (DET) curves can be generated and equal-error rates (EER) can be calculated. These 10x10 matrices of equal-error rates were the common basis in this study for comparison of speaker recognition techniques.

4 RESULTS AND DISCUSSION

Results will be presented in this section for the following experiments which were conducted as part of the research effort:

- A large-scale study of GMM-UBM parameter sensitivity to develop an appropriate baseline model
- Cross-condition EER matrices using the baseline GMM-UBM and the three pattern classification methods applied to the GMM supervector features
- Results of speaker recognition in the PLS subspace using the pattern classification methods, with comparison to the best performing technique that used the GMM supervector based features. These results will be presented for all four of the development data set configurations. The CDDR decomposition technique was not fully developed to the point of implementation in the speaker recognition system, so results will focus solely on the PLS decomposition.
- Effect of the dimensionality of the PLS subspace on performance of the speaker recognition system.

4.1 Baseline GMM-UBM parameter sensitivity

There were two experiments that were conducted to evaluate the sensitivity of the GMM-UBM model to the parameters of the ALIZE/LIA toolbox functions. The first experiment examined the effect of several EnergyDetector function parameters on the number of frames that were retained. Figure 3 plots the correlations between the number of frames in the feature file and each of the EnergyDetector function parameters included in the sensitivity analysis. There are two methods within the EnergyDetector function for setting the threshold for retaining frames: a weighted threshold based on the component weightings in the GMM, and a mean-based threshold that is calculated by subtracting a multiple of the standard deviation from the component mean (i.e. “meanStd”). The correlations plotted in Figure 3 clearly show that, for each threshold method in EnergyDetector, there is only a single relevant parameter controlling the number of selected frames. The “weighted” threshold method in Energy Detector was used throughout this research effort, so a follow-up examination of the number of GMM components (“mixDistribCount”) was conducted.

The second phase of the sensitivity analysis focused on the two most relevant parameters in the generation of the GMM-UBM model: the number of components in the GMMs in both Energy Detector and the GMM-UBM model. Cross-condition testing on all ten MultiRoom8 data sets results in a set of 100 equal-error rates (EERs). Figure 4 shows the median EER for cross-condition training and testing when using four different values for the number of GMM-UBM model components and five values for the number of GMM components in the energy detector (20 pairs in total). The left subplot shows median EER over a set of 100 speaker recognition experiments; the right subplot shows the increase in EER over the minimum EER value from the left subplot. Darker colors indicate lower EERs and better performance conditions. The strongest trend is the poor performance when using only two components in the Energy Detector

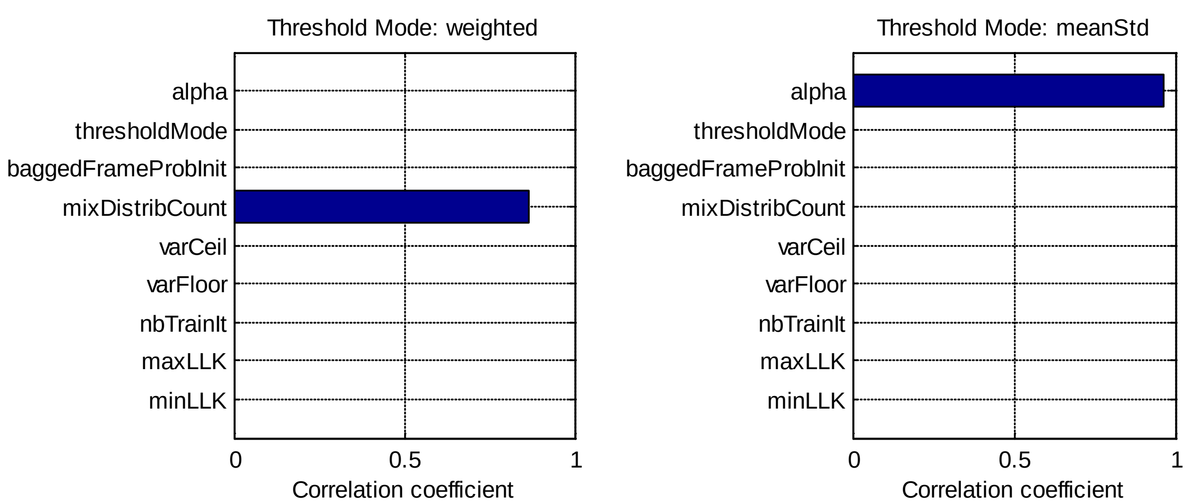

Figure 3. Correlation between the values of the “EnergyDetector” function parameters and the size (i.e. number of frames) of the feature file for each recording.

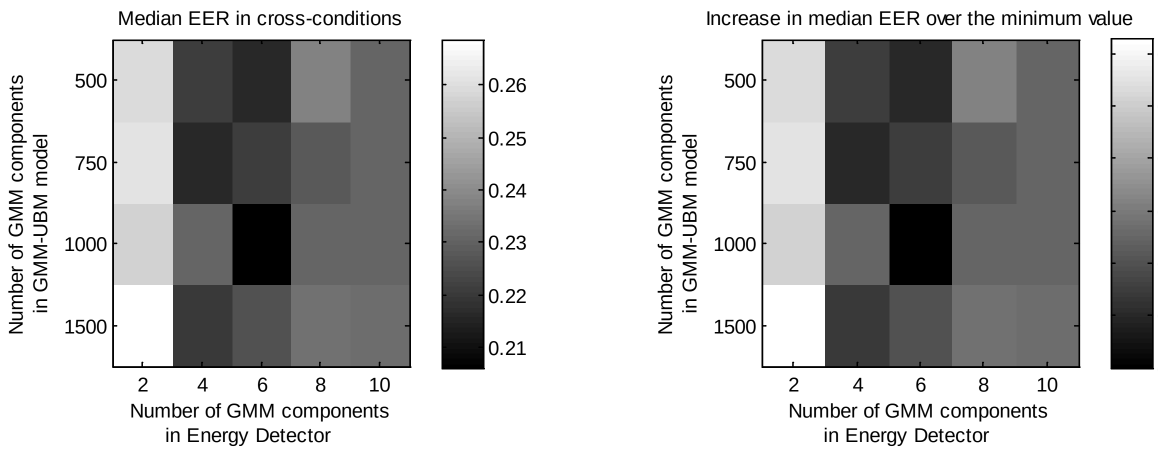

Figure 4. Representation of the equal-error rates (EER) as a function of two parameters in the GMM-UBM model: the number of components used in the GMM in the EnergyDetector function (x-axis) and the number of components in the GMM used in TrainTarget to learn the distribution of MFCCs (y-axis).

GMM (leftmost column of left subplot); performance is also slightly worse than the best – performing value when using a larger number (8 to 10) of components in the Energy Detector GMM. Best overall performance is achieved when using six components in the Energy Detector GMM and a 1000-component GMM-UBM. Given the likelihood that the parameter values corresponding to the overall minimum in EER are benefiting from some overfitting (and the result would not be consistent in validation with a new test set), the GMM-UBM model in this research effort was constructed using six components in the Energy Detector GMM and a 750-component GMM-UBM model. The median EER for these parameter setting is 1.5% higher (20.6% versus 22.1%) than the overall best parameter settings.

4.2 MultiRoom8 speaker recognition results

Baseline GMM-UBM speaker recognition results for the MultiRoom8 cross-condition training and testing are shown in Table 5. The table contains the equal-error rates (EERs) for each training and test condition. As expected, the table indicates substantial degradation in performance (i.e. increase in EER) when there is a mismatch between the training and testing conditions. This can be seen by comparing the values on the diagonal (same condition train and test) with off-diagonal values in the same column. The average increase in off-diagonal EER in each column versus "same condition" EER is 10.4%. One interesting exception to this trend is when training in the "Large, Dir@3ft" condition. Due to the noise conditions present in the large room, performance often improves when the test data is from a "better" condition, even if it results in a mismatch in the training and test conditions.

The first sophistication beyond the baseline GMM-UBM that was considered in the speaker recognition system was to use extracted GMM supervectors as inputs to pattern recognition techniques. Three widely-known pattern recognition algorithms were considered: nearest neighbor, Random Forest, and the support vector machine (SVM). These algorithms have

Table 5. Matrix of equal-error rates (EERs) using the baseline 750-component GMM-UBM for the 100 cross-condition experiment setups constructed with the MultiRoom8 data set.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	3.8%	21.0%	21.0%	28.2%	30.8%	12.0%	22.9%	22.7%	23.3%	38.5%
Small	Dir @ 3ft	25.8%	10.3%	20.5%	28.6%	20.8%	16.2%	15.4%	19.9%	12.8%	31.4%
Medium	Dir @ 3ft	22.7%	22.2%	17.4%	31.1%	20.5%	11.4%	13.4%	13.6%	18.9%	30.6%
Large	Dir @ 3ft	28.2%	28.6%	30.6%	29.4%	24.3%	25.0%	22.9%	27.8%	23.7%	23.6%
Small	Dir @ 5ft	32.7%	22.1%	20.6%	26.9%	7.6%	25.6%	25.6%	20.5%	21.4%	29.7%
Medium	Omni @ close	9.6%	13.2%	7.6%	19.5%	23.8%	2.5%	14.2%	13.6%	12.9%	30.6%
Small	Omni @ mid-dist	23.1%	14.0%	20.5%	22.9%	17.8%	18.0%	10.3%	20.2%	16.1%	33.3%
Medium	Omni @ mid-dist	22.9%	20.6%	13.6%	24.0%	23.1%	13.0%	20.5%	6.8%	17.9%	30.6%
Small	Omni @ far	25.6%	20.2%	25.6%	27.2%	19.1%	23.1%	18.0%	20.5%	11.6%	33.2%
Large	Omni @ far	33.7%	36.9%	36.1%	28.6%	35.1%	36.1%	35.6%	30.6%	32.2%	27.7%

characteristics that make them particularly suitable for straightforward application to the GMM supervector speaker recognition task: they can estimate any necessary parameters with a single training vector per speaker and they can manage the high-dimensional feature space. The SVM applied to GMM supervectors has been used in several studies of speaker recognition and NIST evaluations, and is widely viewed as a preferable approach when compared to the classic GMM-UBM.

The equal-error rates for all cross-condition training and testing using the GMM Supervector Nearest Neighbor (GMMSV-NN) are shown in Table 6. The difference between the EERs using the GMMSV-NN and the baseline GMM-UBM are shown in Table 7, with positive values indicating better performance (lower EER) with the GMMSV-NN method. The color coding in the table indicates changes in EER of at least 5%, with green cells indicating better performance with the GMMSV-NN and red cells indicating better performance with the baseline GMM-UBM. Overall, the GMMSV-NN does not appear to improve upon the GMM-UBM baseline; instead, there is some degradation in performance, particularly in mismatched training and test conditions. Several of the EERs that did improve are for same-condition train and test, which further widens the gap between matched-condition and mismatched-condition equal-error rates. For the GMMSV-NN, the average penalty for mismatched training and test conditions (i.e. off-diagonal EERs) relative to EER for same-condition training and testing (i.e. EERs on the diagonal) is 20.6%. One potential influence in this large increase is that the “Large, Dir@3ft” same-condition training and test EER was substantially improved with the GMMSV-NN, such that this column in the EER matrix is now also deleteriously contributing to the average penalty for mismatched conditions.

The next classification technique to be applied to the GMM supervector features was the Random Forest classifier. Table 8 shows the EER matrix, and Table 9 shows the change in EER

from the GMM-UBM baseline with the same color coding as Table 7, with positive values indicating better performance with the GMMSV-RF. Performance with the GMMSV-RF is universally degraded, with increases in all EERs in the cross-condition train and test matrix. While the Random Forest should benefit from the high-dimensionality of the feature space (since it decreases correlation between the individual decision trees), the Random Forest classifier can be impacted by the presence of a significant number of uninformative features. These decision trees, constructed largely from noisy features, can overwhelm and outnumber the smaller percentage of component decision trees that would correctly classify the test sample.

The final classification methods considered in this study was the Support Vector Machine (SVM), which has been included in many previous investigations of speaker recognition (e.g. [11]). In this experiment, the SVM with a linear kernel was used to develop a classifier for discriminating between each possible pair of speakers. The EERs for the cross-condition testing are shown in Table 10, and the change in EER versus the baseline GMM-UBM (with green and red color coding of improvements and degradations) is shown in Table 11. The large number of green-shaded cells in Table 11 indicates that the GMMSV-SVM provides improved performance for many of the training and testing conditions. The average penalty for mismatched training and test conditions is 16.3%, which is still higher than the GMM-UBM value, indicating that a greater improvement is seen in the matched conditions than in the mismatched conditions.

Thus, amongst the techniques operating on the GMM supervector, the GMMSV-SVM provides the best performance on the MultiRoom8 data set as well as outperforming the GMM-UBM baseline by a substantial margin in many conditions. This result is consistent with the research literature regarding performance of the GMMSV-SVM in speaker recognition tasks. An analysis of patterns within the performance of the GMM-SVM reveals no significant preferences for certain conditions or scenarios within the results. Similarly, the level of improvement over the GMM-UBM baseline does not appear to be influenced by the room or microphone conditions. Figure 5 shows a two-dimensional representation of the results in Table 10. Each point in the graph represents one of the ten MultiRoom8 conditions, and distances between points are calculated from the EER matrix (with EER serving as a proxy for distance). Also included in the figure are four additional conditions that used significantly different recording devices (GSM and CDMA cellphones, landline, and push-to-talk radio). Low equal-error rates would result in points close together, and larger equal-error rates will force points to be further apart. The lack of clusters and approximately equal spacing between points representing the omnidirectional and directional microphones indicates the lack of strong preference within the GMM-SVM framework for a particular experiment setup; similarities in microphone, room, or recording distance do not result in significantly tighter clusters of points.

Table 6. Matrix of equal-error rates (EERs) using the nearest neighbor classifier applied to the GMM supervector (GMMSV-NN) for the 100 MultiRoom8 cross-condition experiment setups.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	5.1%	28.2%	28.1%	30.2%	28.7%	10.4%	28.1%	31.8%	30.2%	41.0%
Small	Dir @ 3ft	37.1%	10.3%	23.1%	25.7%	25.6%	28.3%	25.6%	25.6%	23.1%	40.0%
Medium	Dir @ 3ft	33.3%	19.7%	5.3%	24.7%	30.8%	25.0%	27.7%	25.0%	28.2%	43.3%
Large	Dir @ 3ft	38.5%	23.2%	19.5%	7.7%	27.9%	34.2%	34.3%	29.7%	33.7%	33.3%
Small	Dir @ 5ft	38.6%	27.1%	33.3%	32.4%	9.5%	30.8%	28.0%	28.4%	18.7%	35.9%
Medium	Omni @ close	11.7%	23.9%	19.9%	24.2%	29.3%	2.3%	20.1%	20.5%	23.0%	41.7%
Small	Omni @ mid-dist	35.9%	20.5%	23.9%	34.3%	21.8%	24.6%	8.2%	23.1%	10.3%	31.4%
Medium	Omni @ mid-dist	31.8%	25.0%	16.5%	22.2%	28.2%	14.5%	26.1%	4.5%	25.9%	33.3%
Small	Omni @ far	33.8%	25.6%	28.2%	31.6%	23.8%	30.8%	19.0%	29.8%	5.7%	39.4%
Large	Omni @ far	44.5%	40.0%	38.9%	31.9%	34.1%	41.7%	40.0%	36.1%	36.8%	22.1%

Table 7. Differences between the EER matrices for the GMMSV-NN and the baseline GMM-UBM. Positive values indicate better performance with the GMMSV-NN. Cells shaded green identify changes in EER of at least 5%; cells shaded red identify changes in EER of at least -5%.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	-1.2%	-7.2%	-7.1%	-2.0%	2.1%	1.6%	-5.1%	-9.1%	-7.0%	-2.6%
Small	Dir @ 3ft	-11.4%	0.0%	-2.6%	2.9%	-4.9%	-12.1%	-10.3%	-5.7%	-10.3%	-8.6%
Medium	Dir @ 3ft	-10.6%	2.5%	12.1%	6.4%	-10.3%	-13.6%	-14.3%	-11.4%	-9.3%	-12.7%
Large	Dir @ 3ft	-10.3%	5.4%	11.1%	21.7%	-3.6%	-9.2%	-11.5%	-1.9%	-10.0%	-9.7%
Small	Dir @ 5ft	-5.9%	-5.1%	-12.8%	-5.6%	-1.9%	-5.1%	-2.4%	-7.9%	2.7%	-6.2%
Medium	Omni @ close	-2.1%	-10.7%	-12.3%	-4.7%	-5.5%	0.2%	-5.9%	-6.8%	-10.1%	-11.1%
Small	Omni @ mid-dist	-12.8%	-6.5%	-3.4%	-11.4%	-4.0%	-6.6%	2.0%	-2.9%	5.9%	1.9%
Medium	Omni @ mid-dist	-8.9%	-4.4%	-2.9%	1.8%	-5.1%	-1.5%	-5.6%	2.3%	-8.0%	-2.8%
Small	Omni @ far	-8.2%	-5.4%	-2.6%	-4.4%	-4.8%	-7.7%	-1.0%	-9.3%	5.9%	-6.2%
Large	Omni @ far	-10.8%	-3.1%	-2.8%	-3.2%	1.0%	-5.6%	-4.4%	-5.6%	-4.7%	5.6%

Table 8. Matrix of equal-error rates (EERs) using the Random Forest classifier applied to the GMM supervector (GMMSV-RF) for the 100 MultiRoom8 cross-condition experiment setups.

Training condition		Omni @ close	Dir @ 3ft	Dir @ 3ft	Dir @ 3ft	Dir @ 5ft	Omni @ close	Omni @ mid-dist	Omni @ mid-dist	Omni @ far	Omni @ far
Conf	Omni @ close	9.9%	39.6%	28.6%	33.7%	41.8%	25.6%	40.9%	39.1%	42.5%	45.0%
Small	Dir @ 3ft	32.8%	18.8%	28.2%	34.3%	32.2%	34.6%	31.2%	32.4%	35.8%	40.8%
Medium	Dir @ 3ft	34.1%	25.8%	18.0%	31.9%	34.0%	26.0%	38.2%	27.5%	35.3%	50.3%
Large	Dir @ 3ft	37.5%	34.1%	35.6%	20.8%	32.8%	34.7%	39.3%	38.2%	41.0%	47.4%
Small	Dir @ 5ft	45.8%	28.0%	36.8%	45.0%	26.2%	43.5%	32.3%	38.8%	40.2%	42.3%
Medium	Omni @ close	29.7%	32.0%	30.7%	36.7%	47.5%	18.2%	36.0%	26.2%	37.5%	47.1%
Small	Omni @ mid-dist	40.6%	35.1%	33.0%	48.0%	42.2%	34.4%	26.4%	36.1%	33.2%	43.1%
Medium	Omni @ mid-dist	40.0%	41.7%	32.4%	37.6%	35.5%	35.1%	43.2%	24.4%	37.4%	38.3%
Small	Omni @ far	40.4%	32.5%	38.1%	43.1%	43.1%	35.8%	31.1%	38.8%	28.1%	42.2%
Large	Omni @ far	45.7%	47.4%	37.4%	42.3%	48.1%	39.4%	39.2%	49.9%	40.4%	38.2%

Table 9. Differences in the equal-error rates between the GMMSV-RF and the baseline GMM-UBM.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	-6.0%	-18.6%	-7.6%	-5.5%	-11.0%	-13.6%	-18.0%	-16.4%	-19.2%	-6.5%
Small	Dir @ 3ft	-7.0%	-8.6%	-7.7%	-5.7%	-11.4%	-18.4%	-15.8%	-12.5%	-23.0%	-9.4%
Medium	Dir @ 3ft	-11.4%	-3.5%	-0.6%	-0.8%	-13.5%	-14.6%	-24.8%	-13.9%	-16.4%	-19.8%
Large	Dir @ 3ft	-9.3%	-5.5%	-5.0%	8.6%	-8.5%	-9.7%	-16.4%	-10.4%	-17.3%	-23.8%
Small	Dir @ 5ft	-13.2%	-5.9%	-16.3%	-18.1%	-18.6%	-17.8%	-6.6%	-18.3%	-18.7%	-12.6%
Medium	Omni @ close	-20.1%	-18.8%	-23.1%	-17.3%	-23.7%	-15.7%	-21.9%	-12.6%	-24.6%	-16.5%
Small	Omni @ mid-dist	-17.6%	-21.1%	-12.5%	-25.2%	-24.4%	-16.5%	-16.1%	-15.8%	-17.1%	-9.8%
Medium	Omni @ mid-dist	-17.1%	-21.2%	-18.8%	-13.6%	-12.5%	-22.1%	-22.7%	-17.6%	-19.6%	-7.7%
Small	Omni @ far	-14.8%	-12.3%	-12.5%	-16.0%	-24.1%	-12.7%	-13.2%	-18.3%	-16.5%	-9.0%
Large	Omni @ far	-12.0%	-10.5%	-1.3%	-13.6%	-12.9%	-3.3%	-3.6%	-19.3%	-8.3%	-10.5%

Table 10. Matrix of equal-error rates (EERs) using the linear kernel support vector machine applied to the GMM supervector (GMMSV-SVM) for the 100 MultiRoom8 cross-condition experiment setups.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	1.3%	16.4%	15.4%	23.1%	26.4%	4.6%	20.9%	20.5%	16.3%	38.5%
Small	Dir @ 3ft	23.1%	0.7%	8.5%	17.1%	10.4%	18.0%	12.8%	18.0%	12.1%	28.6%
Medium	Dir @ 3ft	20.5%	10.1%	0.3%	11.1%	18.0%	13.1%	15.4%	11.7%	13.8%	33.3%
Large	Dir @ 3ft	26.7%	17.1%	11.2%	3.2%	24.3%	19.5%	20.0%	17.4%	25.7%	33.9%
Small	Dir @ 5ft	31.0%	15.4%	20.5%	16.7%	1.6%	25.6%	18.0%	20.5%	16.4%	32.4%
Medium	Omni @ close	6.8%	12.8%	4.6%	14.5%	22.3%	0.2%	15.4%	6.9%	12.8%	36.1%
Small	Omni @ mid-dist	23.7%	10.3%	15.4%	21.4%	15.4%	18.0%	5.1%	20.3%	5.1%	27.7%
Medium	Omni @ mid-dist	22.7%	9.8%	8.3%	19.5%	18.5%	8.4%	19.0%	1.0%	20.5%	30.6%
Small	Omni @ far	25.6%	12.8%	20.2%	20.4%	19.1%	18.0%	10.3%	16.5%	2.6%	36.8%
Large	Omni @ far	33.0%	24.5%	20.3%	21.7%	27.0%	29.1%	28.6%	21.4%	26.3%	12.8%

Table 11. Differences in the equal-error rates between the GMMSV-SVM and the baseline GMM-UBM.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	2.6%	4.7%	5.6%	5.1%	4.4%	7.4%	2.0%	2.3%	7.0%	0.0%
Small	Dir @ 3ft	2.7%	9.5%	12.0%	11.4%	10.3%	-1.7%	2.6%	2.0%	0.7%	2.9%
Medium	Dir @ 3ft	2.3%	12.2%	17.1%	20.0%	2.6%	-1.7%	-2.0%	2.0%	5.1%	-2.8%
Large	Dir @ 3ft	1.5%	11.4%	19.4%	26.2%	0.0%	5.6%	2.9%	10.4%	-2.0%	-10.3%
Small	Dir @ 5ft	1.7%	6.7%	0.1%	10.1%	6.0%	0.0%	7.7%	0.0%	5.0%	-2.7%
Medium	Omni @ close	2.8%	0.4%	3.1%	4.9%	1.5%	2.3%	-1.2%	6.8%	0.1%	-5.6%
Small	Omni @ mid-dist	-0.7%	3.8%	5.1%	1.5%	2.4%	0.0%	5.1%	-0.1%	11.1%	5.7%
Medium	Omni @ mid-dist	0.2%	10.7%	5.4%	4.5%	4.6%	4.7%	1.6%	5.9%	-2.6%	0.0%
Small	Omni @ far	0.0%	7.4%	5.4%	6.8%	0.0%	5.1%	7.7%	4.1%	9.0%	-3.7%
Large	Omni @ far	0.7%	12.4%	15.8%	6.9%	8.1%	7.1%	7.0%	9.2%	5.8%	14.9%

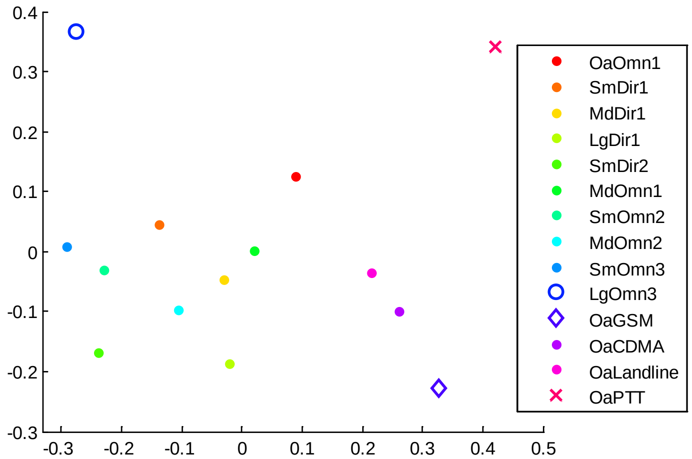

Figure 5. Representation of the distance between the MultiRoom8 conditions based on equal-error rates for the GMMSV-SVM.

Figure 6 shows a scatter plot comparing the EERs between the baseline GMM-UBM and the GMMSV-SVM. Each point represents one of the entries in the EER matrices (thus, there are a total of 100 points). The red and green lines in the plot indicate the +/- 5% thresholds for the red and green coding shown in Table 10; points above the green line would be shaded green and points below the red line would be shaded red. The conditions for which performance improves when using the GMM-SVM are not concentrated at any particular level of baseline GMM-UBM performance; the GMM-SVM improves performance in many conditions where the baseline GMM-UBM did relatively poorly and also conditions where it did relatively well (i.e. better than its average).

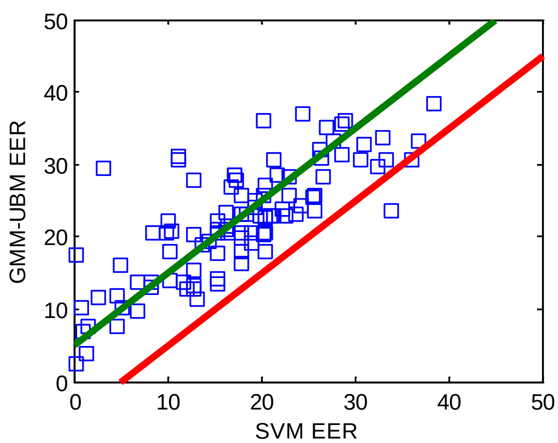

Figure 6. Scatter plot of equal-error rates for the GMMSV-SVM and baseline GMM-UBM on the 100 MultiRoom8 cross-condition experiment configurations.

4.3 Effect of supervector decomposition on speaker recognition system performance

The speaker recognition results using supervector decomposition are dependent on the composition of the development data set used to estimate the subspace projection coefficients. Four possible development data sets were considered, with each development data set comprised of files from various speakers and conditions (with none of the training or testing files ever appear in the development data set). Recall that the train and test sets only consisted of ten speakers, which were the first ten speakers when organized by speaker ID.

The first development data set was compiled using all speakers (including the ten speakers in the training and test sets) but in different room/microphone conditions than used for training and testing. The GMM supervector was projected into a 25-dimensional feature space using PLS, and the three pattern classification techniques (nearest neighbor, Random Forest, and support vector machine) were applied. Figure 7 through Figure 9 show the results using nearest neighbor, SVM, and Random Forest, respectively. Each figure includes four subplots. In the top row of subplots, the classifier post-PLS projection is compared to the classifier applied to the high-dimensional GMM supervector. The scatter plot shows the EERs for each classifier setup for each of the 100 cross-conditions (points above the solid diagonal line indicate a reduction in EER and benefit from using PLS decomposition). The histogram on the right shows the distribution of improvements in EER, where positive values indicate a reduction in EER and better performance using the PLS decomposition. The bottom row of subplots in Figure 7 through Figure 9 show each classifier post-PLS compared to the GMMSV-SVM, which was the best performing technique when operating on the raw GMM supervector.

In Figure 7, the nearest neighbor classifier applied to the PLS-decomposed supervectors (PLS-NN) substantially outperforms the GMMSV-NN. The median improvement is 11.9%. Similarly, and perhaps more significant, the PLS-NN also outperforms the GMMSV-SVM with a median improvement of 11.5% in the 100 cross-condition speaker recognition tasks. In Figure 8, the SVM with a radial basis function kernel is applied to the PLS-decomposed supervectors (PLS-SVM). The SVM with a radial basis function kernel was not well-suited for classification in the high-dimensional GMM supervector space; thus the SVM in the upper left subplot of Figure 8 is performing at near chance (50% EER) for all 100 cross-conditions. In the lower pair of subplots in Figure 8, the PLS-SVM is compared with the linear kernel SVM applied to the GMM supervector (GMMSV-SVM). In this comparison, the PLS-SVM is able to provide a median improvement of 7.1%. In Figure 9, the Random Forest classifier was applied to the PLS-decomposed supervectors (PLS-RF). As discussed previously regarding Table 8, the Random Forest classifier did not perform well in the high-dimensional supervector feature space. Thus, the Random Forest classifier in the upper left subplot is exhibiting near chance performance. When the PLS-RF classifier is compared to the GMMSV-SVM, the differences are more evenly split in a bimodal distribution centered at zero. However, the median change in EER is still 5.1% (a net improvement).

These results suggest that significant improvements in the median EER could be achieved using PLS projections developed from all speakers (including the ten speakers in the training and test sets) recorded in different room/microphone conditions. The result is a valid experiment design; however it is unlikely that this type of development data will be available in many scenarios. Since the MultiRoom8 data set contains 10 conditions, and only two will be used for training and testing on any given iteration, there will be eight recordings of each test-set speaker in the development data set (albeit in different recording conditions). Thus, the PLS subspace projection gets the opportunity to learn mappings similar to those illustrated in Figure 2 for the speakers that are present in the test set.

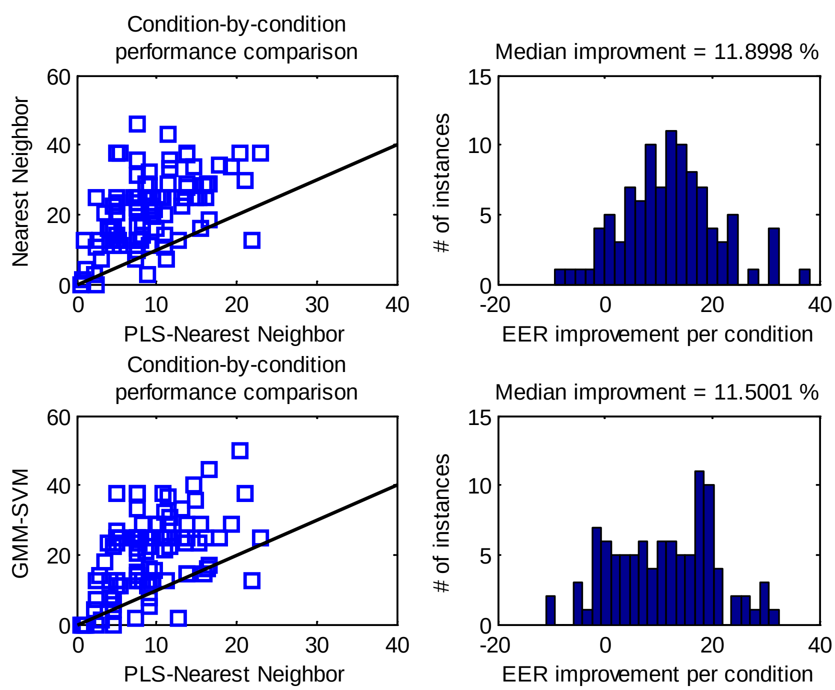

Figure 7. Performance of the nearest neighbor classifier with PLS projection using the first development data set.

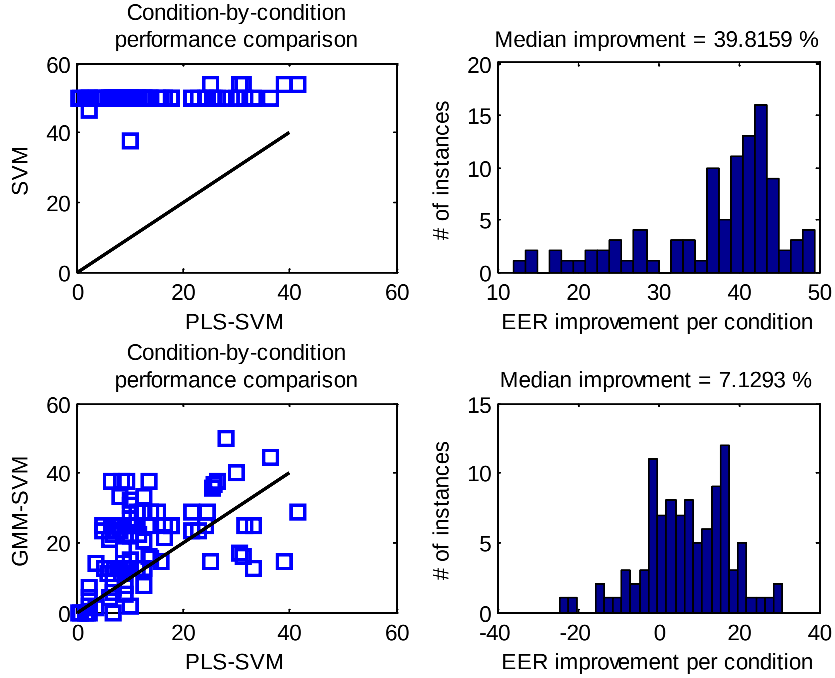

Figure 8. Performance of the radial-basis kernel SVM classifier with PLS projection using the first development data set.

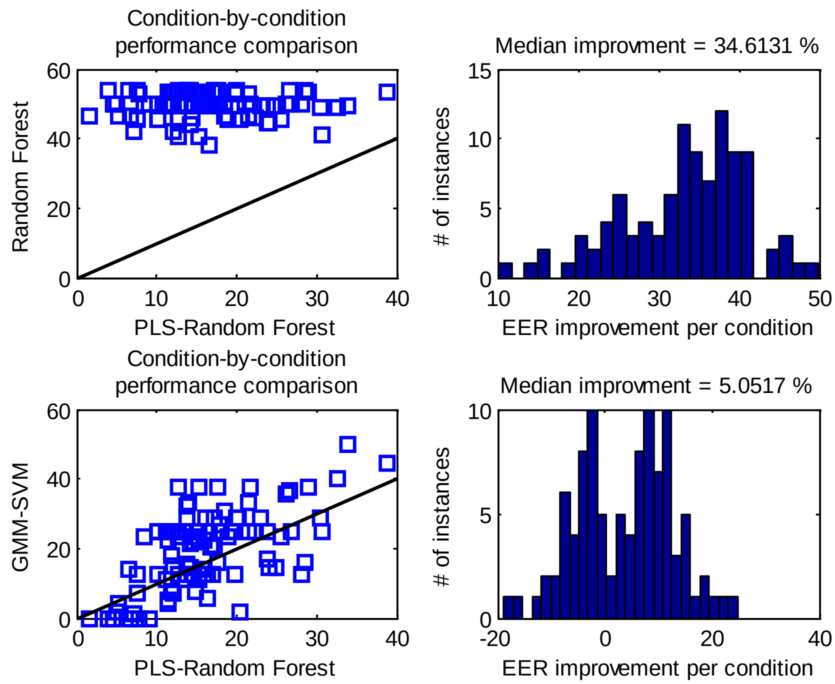

Figure 9. Performance of the Random Forest classifier with PLS projection using the first development data set.

The second development data set was compiled using the same conditions as the training and test data but excluding the ten speakers in the test data set (i.e. same conditions, different speakers). Thus, the PLS projection will be learned from recordings in environments that are most relevant to the recognition task, but for different speakers than those in the test set. From the plots in Figure 10, it can be seen that PLS-NN improves over both the GMMSV-NN and the GMMSV-SVM, although the median improvement is much less than what was observed with the prior development data set. In Figure 11 and Figure 12, it can be seen that the radial-basis kernel SVM and Random Forest classifiers do not perform well on the supervector features (these are the same results as shown in Figure 8 and Figure 9 since the raw GMM supervector features are not affected by the development data set). There is also no median improvement when either the PLS-SVM or PLS-RF are compared to the GMMSV-SVM. Thus, the same-condition/different-speakers development data set appears to not contain enough information for the PLS supervector decomposition to learn a mapping that improves performance. This development data set is the smallest of the four considered, which may be one factor affecting the lack of benefit from the PLS decomposition.

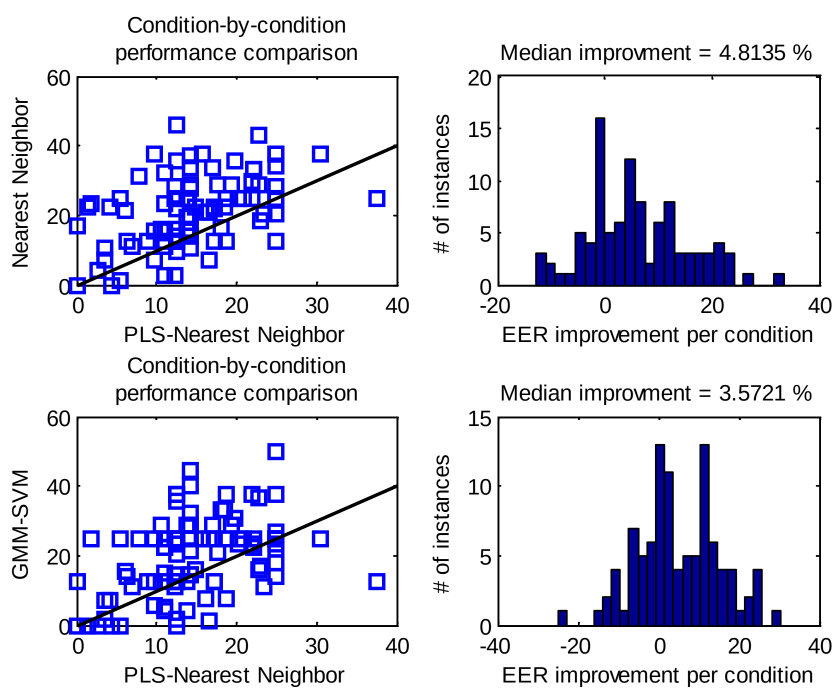

Figure 10. Performance of the nearest neighbor classifier with PLS projection using the second development data set.

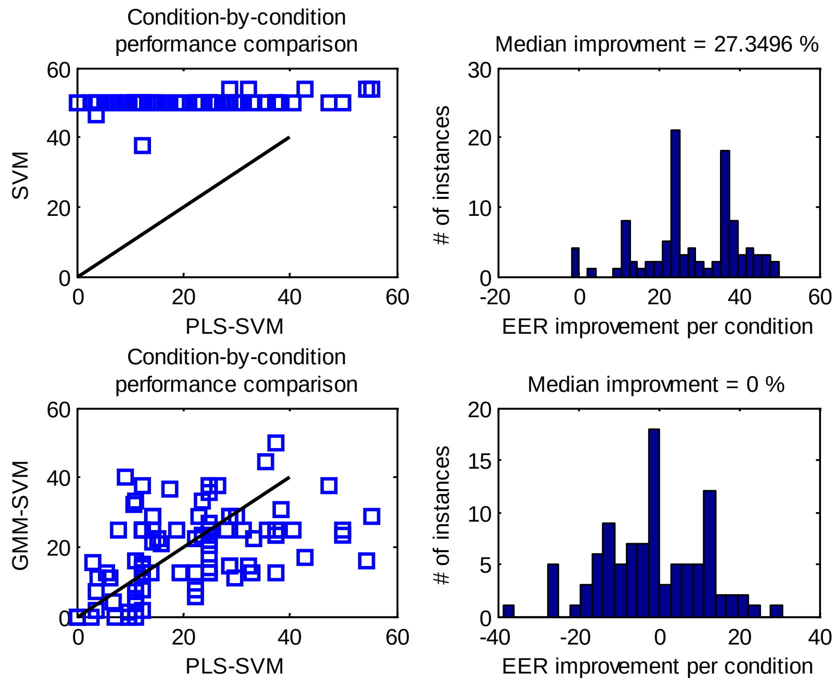

Figure 11. Performance of the radial-basis kernel SVM classifier with PLS projection using the second development data set.

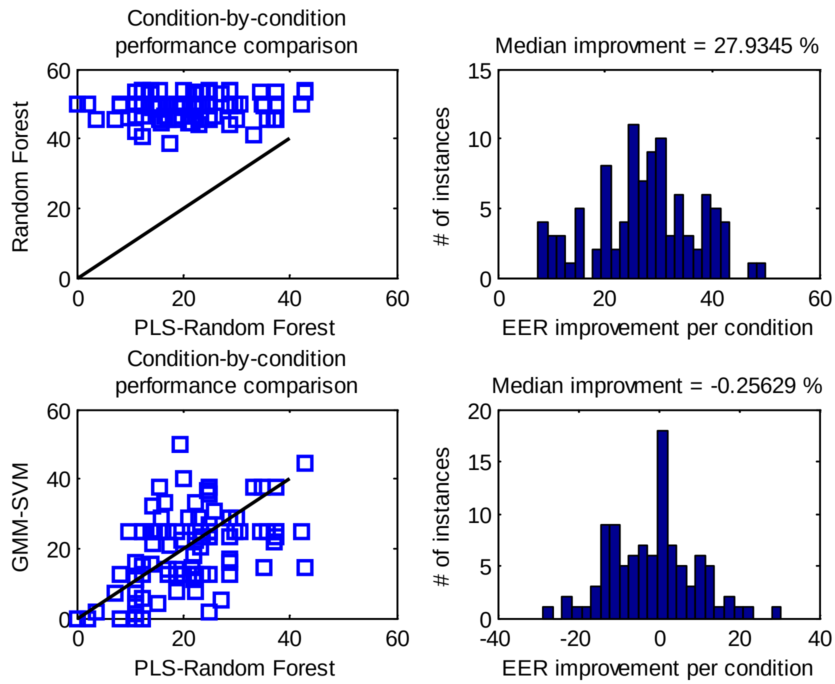

Figure 12. Performance of the Random Forest classifier with PLS projection using the second development data set.

The next development data set consisted of the non-test-set speakers with recordings from all conditions, including training and testing (i.e. all conditions, different speakers). There is no overlap in the test and development speakers, but there is overlap in the test and development conditions. The scatter plots for PLS-NN in Figure 13 show improved EERs for a number of cross-conditions, with a median improvement in EER of 7.0% over the GMMSV-NN and an improvement of 3.6% over the GMMSV-SVM. Thus, for a more realistic scenario of development data (i.e. large amounts of development data, from conditions including but not limited to the training and test conditions, with different speakers), the PLS decomposition is able to improve upon the SVM operating on the GMM supervector. In Figure 14, the radial-basis kernel SVM is also able to improve upon the linear-kernel SVM operating on the GMM supervector, with a median improvement of 2.0%. However, the Random Forest classifier again failed to improve upon the GMMSV-SVM baseline as indicated by the results shown in Figure 15. Since the PLS projection reduces the dimensionality of the feature space to $D = 25$, the Random Forest should not be suffering from the same "noise feature" impairment observed when the Random Forest is applied to the raw GMM supervectors. However, in this situation, it is possible that the Random Forest is limited by the small number of training samples (only ten speakers).

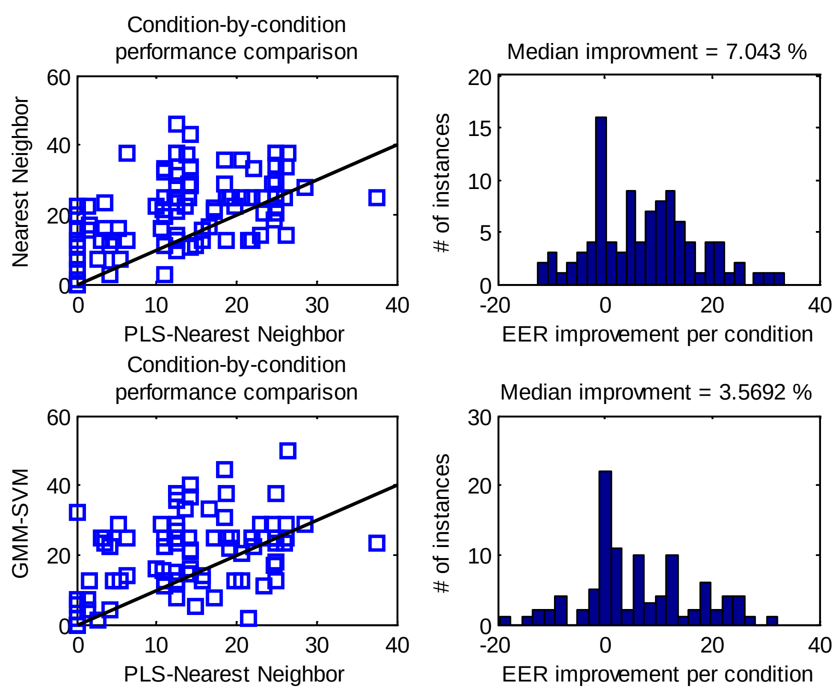

Figure 13. Performance of the nearest neighbor classifier with PLS projection using the third development data set.

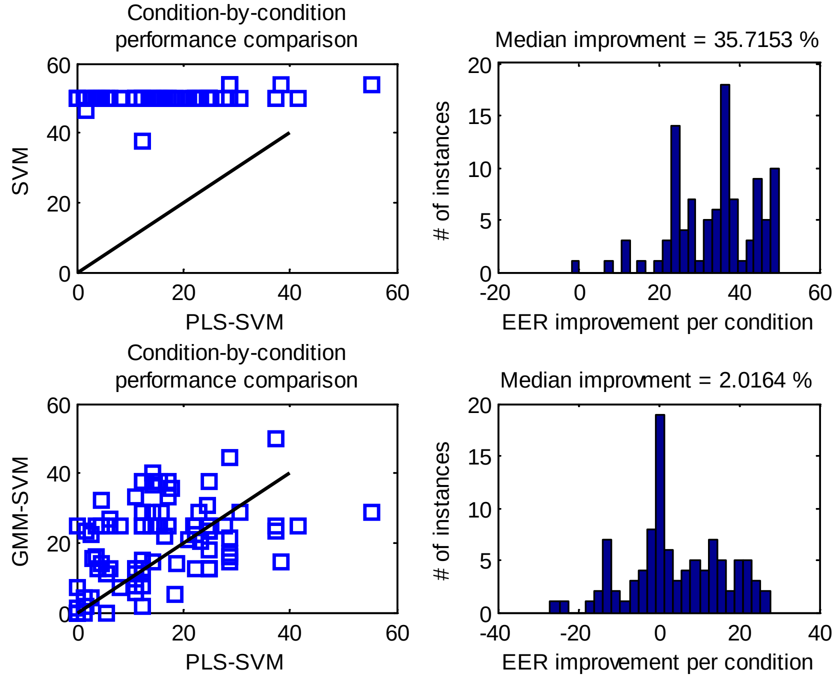

Figure 14. Performance of the radial-basis kernel SVM classifier with PLS projection using the third development data set.

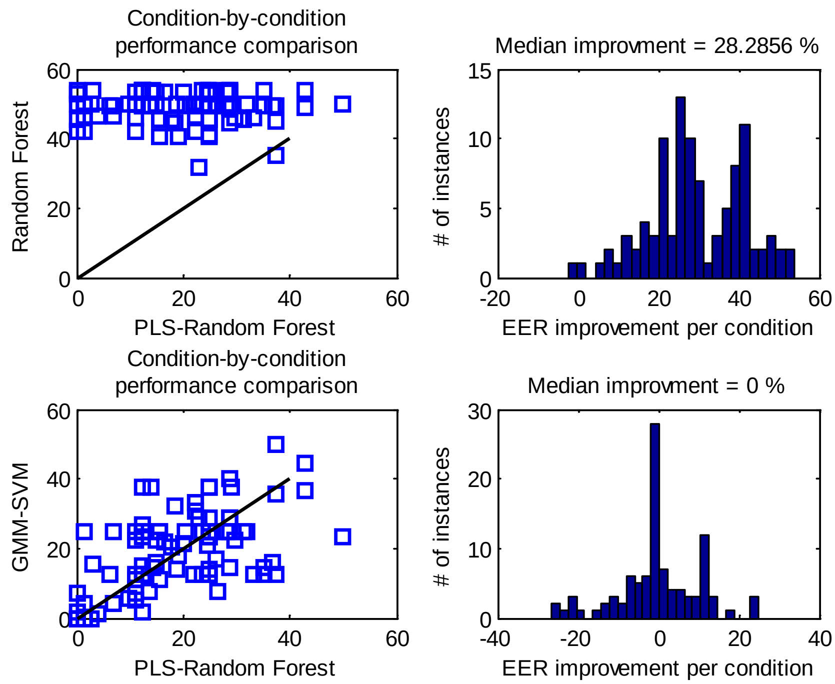

Figure 15. Performance of the Random Forest classifier with PLS projection using the third development data set.

The final development data set was compiled using different speakers and different conditions than those used for training and testing (i.e. different conditions, different speakers). Thus, this might represent a scenario where nothing is known about the training and test environments, preventing collection of development data to match either the training or test conditions. Overall, improvements in EER with the classifiers applied to the PLS decomposed supervectors are at least as good or better than observed with the "different speakers, all conditions" development data set. In Figure 16, PLS-NN has a median reduction in EER of 4.6% when compared to the GMMSV-SVM. In Figure 17, the radial-basis kernel SVM applied to PLS decomposed supervectors provides a median reduction in EER of 2.7%. However, the Random Forest classifier is again unable to provide a measurable improvement in EER as shown in Figure 18.

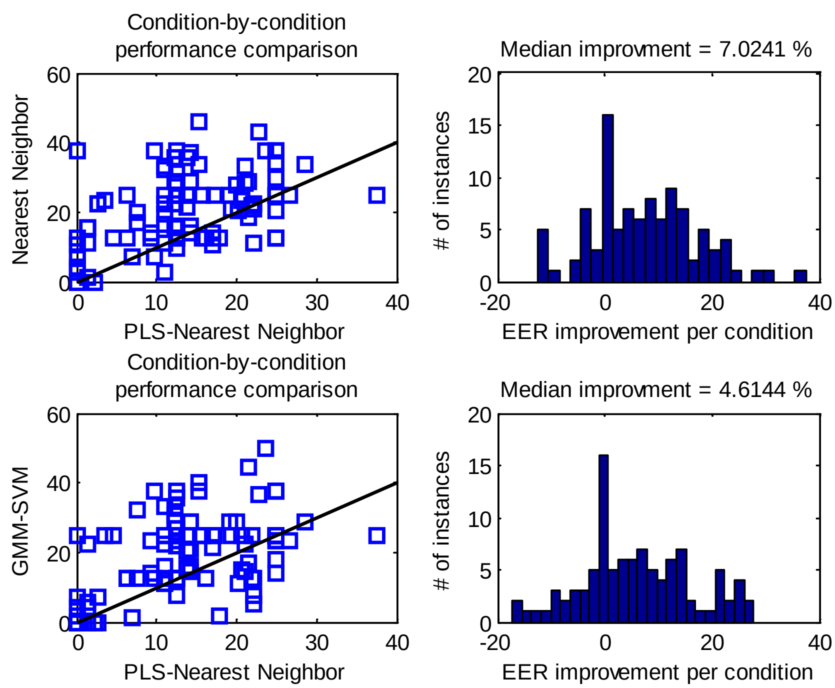

Figure 16. Performance of the nearest neighbor classifier with PLS projection using the fourth development data set.

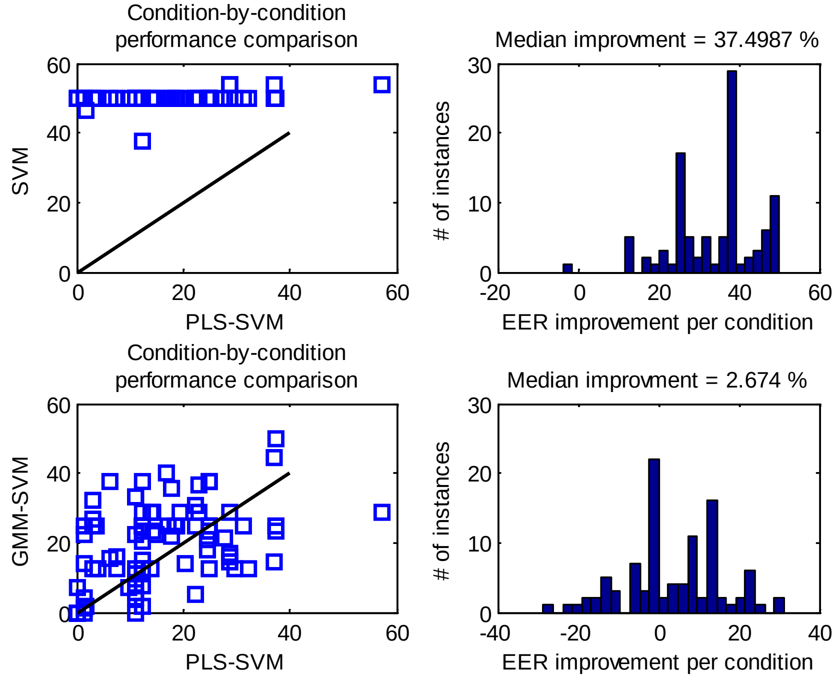

Figure 17. Performance of the radial-basis kernel SVM classifier with PLS projection using the fourth development data set.

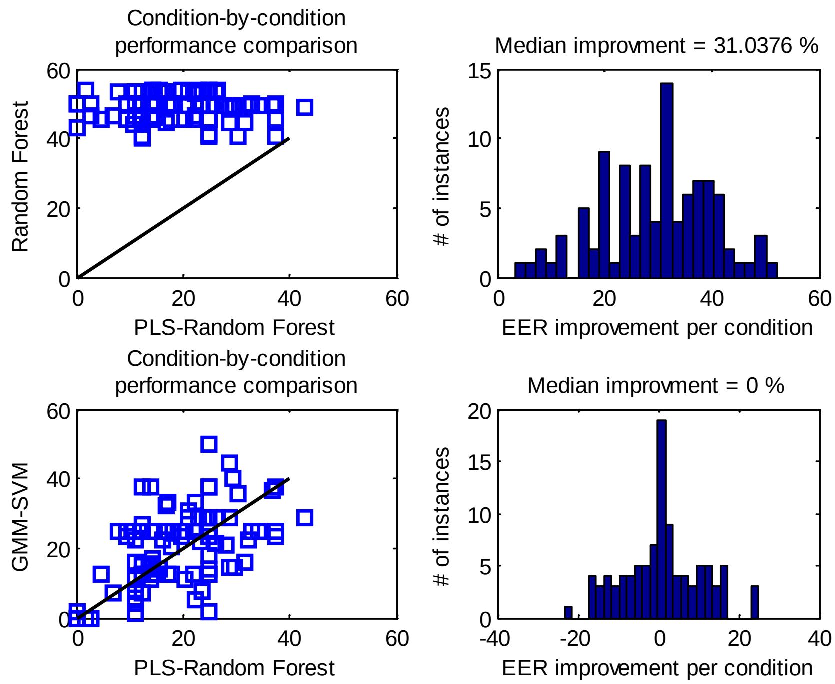

Figure 18. Performance of the Random Forest classifier with PLS projection using the fourth development data set.

Generalizing from the results presented in Figure 7 through Figure 18 for the PLS subspace decomposition, the PLS decomposition paired with the nearest neighbor classifier provided the best EER performance, and also consistently outperformed the GMM supervector SVM. Table 12 is formatted similar to tables presented previously and shows the change in EER when comparing the PLS-NN to the GMMSV-SVM. Positive values indicate better performance (lower EER) using PLS-NN, and green-shaded cells identify changes of at least 5%. The median reduction in EER using PLS-NN is 4.6%, which is 22% of the median EER observed over all 100 cross-conditions with the GMMSV-SVM classifier. Thus, the PLS-NN classifier, when provided with sufficient development data from different speakers in different conditions, was able to reduce EERs by 22%.

Another result of potential interest is direct comparison of the PLS-NN and PLS- radial basis SVM. Both approaches generally showed improvement over the GMMSV-SVM when given appropriate development data. However, it is worth investigating more closely whether the median improvement in EER was a result of each technique performing better on a different subset of the 100 possible cross-condition speaker recognition tasks. If consistent patterns were observed in the EER improvements for the different techniques, and if a relationship between the training/test conditions and the improvement in EER could be learned, it would provide an opportunity for fusion of the two classification methods. Thus, in Table 13 the difference in EER is shown for the PLS-NN and PLS-SVM classifiers. The results are plotted for the "Different Conditions, Different Speakers" development data set with PLS using a 25-dimensional subspace. The green-shaded cells indicate where PLS-NN was capable of an EER at least 5% less than PLS-SVM (negative values in general indicate better performance with PLS-NN). The median change in EER is actually 0%; there are a substantial number of entries

Table 12. Differences in the EER matrices for the GMMSV-SVM and the PLS-NN.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	0.0%	5.6%	-7.4%	7.0%	19.8%	-9.9%	12.5%	-9.9%	0.0%	23.6%
Small	Dir @ 3ft	25.0%	0.0%	-3.1%	4.0%	14.1%	12.5%	21.4%	18.8%	1.6%	22.6%
Medium	Dir @ 3ft	6.2%	0.0%	0.0%	5.5%	16.3%	0.0%	3.1%	18.1%	-7.2%	25.7%
Large	Dir @ 3ft	0.0%	6.8%	10.8%	1.8%	18.8%	4.6%	23.3%	32.3%	10.9%	23.2%
Small	Dir @ 5ft	13.6%	12.5%	0.0%	-1.6%	-1.4%	0.0%	7.8%	-12.5%	1.4%	0.0%
Medium	Omni @ close	0.0%	7.8%	0.0%	7.3%	1.7%	0.0%	-4.7%	5.6%	-8.1%	26.2%
Small	Omni @ mid-dist	25.0%	19.6%	10.9%	0.3%	25.0%	6.3%	7.1%	12.5%	7.8%	18.0%
Medium	Omni @ mid-dist	11.1%	-12.5%	2.8%	-1.5%	2.7%	5.6%	-1.6%	0.0%	-2.7%	0.0%
Small	Omni @ far	12.3%	21.9%	12.2%	3.1%	16.7%	2.6%	12.5%	12.7%	0.0%	12.5%
Large	Omni @ far	0.0%	1.5%	0.4%	0.0%	-14.1%	2.4%	0.0%	-7.7%	0.0%	-19.6%

equal to zero for conditions where both classifiers performed equally. However, the average change in EER is 2.2% (in favor of PLS-NN). The results in Table 13 do not suggest a strong pattern amongst conditions; in fact, the lack of any consistent symmetry in the table may suggest that there is no pattern to which conditions are preferred by one technique or another. In the absence of such patterns, the recommended approach may to be to consider the technique that provides the best average or median performance, which in these experiments was identified to be the PLS-NN method.

One of the primary goals of this effort was to investigate the ability of the subspace decomposition techniques to find a lower-dimensional representation of the GMM supervector that would be less sensitive to changes in the environment and channel. A particular instance that might be illustrative is the experiment setup where the "Conf, Omni @ close" is used for training and "Small, Dir@5ft" is used for testing. These conditions are significantly different, and produce one of the largest EERs in the GMM-UBM baseline. The GMM-UBM baseline achieves an EER of 30.8%, and the GMMSV-SVM reduces the EER to 26.4%. However, using the PLS-NN with a development data set containing different speakers recorded in different conditions and a 25-dimension subspace, the EER can be reduced to 11.1%. Therefore, for at least one example pair of training and test conditions, the PLS subspace decomposition is capable of finding a lower-dimensional feature vector that represents individual speakers with significantly reduced variability or artifacts from the environment.

Table 13. Difference in the EER matrices for PLS-NN and PLS-SVM.

Training condition		Test condition: Conf Omni @ close	Small Dir @ 3ft	Medium Dir @ 3ft	Large Dir @ 3ft	Small Dir @ 5ft	Medium Omni @ close	Small Omni @ mid-dist	Medium Omni @ mid-dist	Small Omni @ far	Large Omni @ far
Conf	Omni @ close	2.2%	1.4%	-7.4%	5.6%	1.2%	0.0%	6.9%	11.1%	6.2%	-13.9%
Small	Dir @ 3ft	-9.7%	0.0%	0.0%	-7.0%	9.4%	0.0%	0.0%	-1.6%	3.1%	0.0%
Medium	Dir @ 3ft	3.7%	1.6%	0.0%	0.0%	6.7%	0.0%	0.0%	0.0%	0.3%	-1.5%
Large	Dir @ 3ft	0.0%	-10.3%	0.0%	-1.8%	0.0%	7.7%	0.3%	4.6%	0.0%	-5.4%
Small	Dir @ 5ft	-11.1%	0.0%	1.2%	0.0%	5.6%	-12.5%	3.1%	9.4%	-2.8%	6.3%
Medium	Omni @ close	0.0%	-6.3%	-8.3%	-10.8%	8.1%	0.0%	0.0%	-6.9%	-12.5%	-15.4%
Small	Omni @ mid-dist	-15.3%	-1.8%	-3.1%	-14.3%	6.3%	0.0%	0.0%	3.1%	7.8%	-10.5%
Medium	Omni @ mid-dist	0.0%	7.8%	-9.7%	-1.5%	-5.3%	-9.7%	4.7%	0.0%	0.0%	-8.6%
Small	Omni @ far	-9.9%	1.6%	-1.5%	-9.4%	0.0%	8.1%	0.0%	1.2%	-1.4%	12.5%
Large	Omni @ far	-12.5%	-14.3%	-22.6%	-16.1%	-10.9%	-28.6%	-7.5%	-7.0%	-12.5%	5.4%

4.4 Effect of supervector decomposition subspace dimensionality

An additional avenue of investigation examined the effect of PLS subspace dimensionality on performance, and potential improvement in EER, for the speaker recognition system. The

results reported previously in Figure 7 through Figure 18 were shown for PLS projections into a 25-dimensional subspace. In Figure 19 through Figure 21, results are shown for all three classifiers (nearest neighbor, SVM, and Random Forest) as a function of the number of PLS subspace dimensions. The boxplots in each figure represent the distribution of EERs within a single cross-condition EER matrix. Red lines are the median of the distribution, upper and lower edges of the blue box identify the 75th and 25th percentile (N = 100), and the red hash symbols indicate outliers that are more than 1.5 standard deviations beyond the edge of the box. Cross-condition EER matrices were generated as the number of PLS subspace dimensions was varied from $D = 5$ to $D = 30$, with the upper limit imposed by the amount of available data. The right side of the plot also shows distributions of EERs for each classifier applied to the high-dimensional GMM supervector, as well as a comparison to the GMMSV-SVM.

There is a consistent trend observed for all three classifiers, and it is particularly noticeable by tracking the median value of the distribution across all of the experiment setups. The most significant improvement is seen as the number of PLS subspace dimensions are increased to 15, beyond which performance continues to improve but with diminishing returns. For all

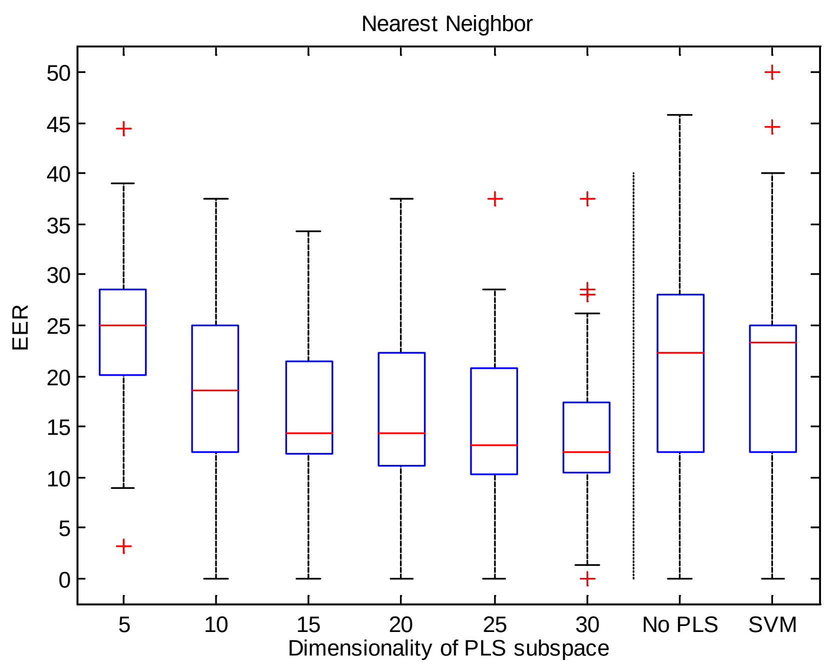

Figure 19. Effect of PLS subspace dimensionality on the distribution of EERs generated using PLS-NN for all 100 MultiRoom8 cross-condition experiment setups.

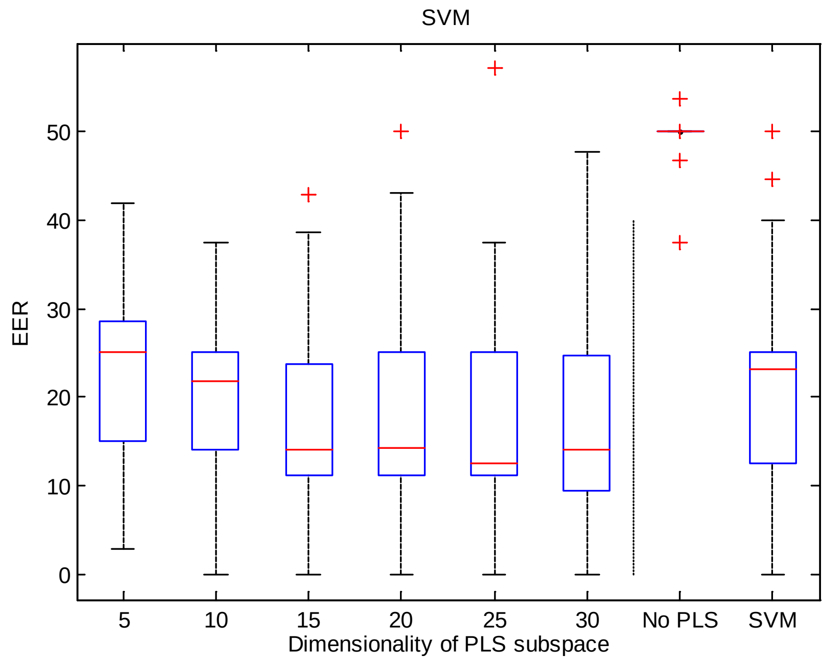

Figure 20. Effect of PLS subspace dimensionality on the distribution of EERs generated using PLS – radial basis kernel SVM for all 100 MultiRoom8 cross-condition experiment setups.

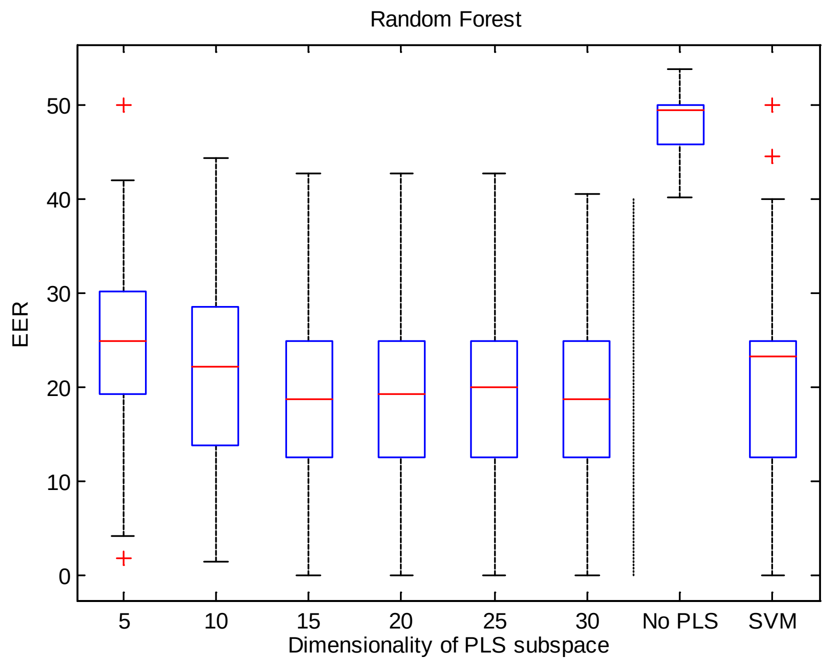

Figure 21. Effect of PLS subspace dimensionality on the distribution of EERs generated using PLS-RF for all 100 MultiRoom8 cross-condition experiment setups.

classifiers, the median value of the EER distribution using a 30-dimensional PLS subspace is lower than the value attained using the GMMSV-SVM. If a larger development data set were available, it would be an interesting study to continue increasing the dimensionality of the PLS subspace to determine at which point the trend fails.

5 CONCLUSIONS AND RECOMMENDATIONS

This technical report describes a set of experiments designed to evaluate various techniques for improving the performance of a speaker recognition system in data conditions that contain both room and microphone variability. Consistent with recent trends in the research community, the primary focus was on dimensionality reduction techniques applied to the GMM supervector, which attempt to find a lower-dimensional subspace that only represents individual speakers and removes the variability introduced by the room and environment. Three pattern classification methods were applied to the decomposed supervectors to determine the most appropriate method for processing the features. The results of the experiments conducted in this research effort provide support for the combination of partial least squares decomposition of the GMM supervector and nearest neighbor classification using a correlation-based distance metric. The combination of these techniques provided significant improvements in equal-error rate when compared to the SVM applied to the GMM supervector, and consistently outperformed both the Random Forest and the SVM applied to the PLS-decomposed supervector.

The development of methods for decomposing supervectors is a topic that is currently receiving significant attention in the speaker recognition research community. The use of partial least squares has several advantages in the application to speaker recognition. The linear projection provides a more tractable and computationally manageable task, particularly given the dimensionality of the GMM supervector. The decomposition is supervised (an advantage over principal component analysis), and finds a subspace projection that optimizes a measure jointly dependent on both the supervectors and the label set. The partial least squares decomposition is also natively capable of handling multiple speakers (i.e. native M-ary classification) and can be run with a single observation per speaker.

The strong performance observed using partial least squares (PLS) on the MultiRoom8 data set motivates further study. A larger data set would enable a more thorough examination of the effects of environment variability. Ideally, a larger data set would contain not only more speaker files (providing more statistical significance to the results) but would also draw from more conditions in the MultiRoom collection such that the impact of a development data set constructed with more complete information can be examined. A data set with greater diversity would also be appropriate for an interesting comparison of more sophisticated constructions of partial least squares. There are several distinct sources of environmental variability in the MultiRoom8 data: different types of microphone (omni vs. directional), distances between the microphone and speaker, and different rooms. There have been efforts to modify the PLS framework in a manner that acknowledges the multidimensional nature of some data collection

environments. The standard PLS framework showed promise in the current research effort; however, these newer PLS methods utilize a more sophisticated construction that may be useful for speaker recognition. Techniques such as Tri-PLS and M-way PLS use a tensor formulation to distinguish the higher-order differences in the data collection conditions, allowing for a more individualized treatment of the sources of variation when constructing the decomposition model. There has been substantial research in the field of chemometrics to develop more sophisticated approaches to PLS decomposition, and these techniques could potentially be useful to the speaker recognition community. Thus, it would be appropriate to consider a cross-disciplinary study of techniques that are being developed for chemometric to the channel and environment variability issues that are currently receiving much attention in speaker recognition.

In addition to the partial least squares approach to supervector decomposition, a technique referred to as classification-directed dimensionality reduction (CDDR) was also considered in this study as a method for nonlinear subspace projection. Recently, a nonlinear extension of PLS (i.e. kernel PLS) has been applied to the speaker recognition task [16]. There is great potential for nonlinear decomposition techniques to outperform simpler, linear projections since the decision to use a linear method is typically due to computational and stability concerns rather than the appropriateness of a linear model. In the consideration of nonlinear subspace decomposition techniques, the challenge is to add sufficient expressivity and flexibility without creating a problem that becomes computationally intractable or ill-posed. The CDDR method has shown promise when applied to other data sets and compared favorably to principal component analysis and partial least squares; unfortunately, efforts with the CDDR method never proceeded beyond the preliminary stage. Further investigation is necessary to evaluate the CDDR method and other nonlinear subspace decompositions in the context of the results presented in this technical report.

The results and conclusions in the current research effort were focused on identifying macro-level trends by comparing performance across conditions. There may be insight to be gained by additional study of the proposed PLS and subspace decomposition techniques with a focus on individual subjects, analyzing performance within Doddington's classic context of sheep, wolves, and goats. Further study of performance for individual types of subjects could potentially motivate strategies for fusion of different methods. For performance on the macro-level, there were not strong patterns identified in the performance of PLS-NN versus PLS-SVM that would clearly indicate a preference for specific methods applied to an entire population for certain conditions. An investigation of the speaker-specific conditions under which specific algorithms may be preferable, or fusion of multiple algorithms, could be a promising avenue for further research that is motivated by the significant differences between the algorithms observed for at least some of the conditions (i.e. there is never a universally best method. Parameterizing the preferences or fusion parameters in terms of the training/testing condition mismatch would be the desired outcome of such an investigation.

6 REFERENCES

[1] B. V. Srinivasan, D. N. Zotkin and R. Duraiswami, "A partial least squares framework for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on,* 2011, pp. 5276-5279.

[2] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field and M. Contolini, "Eigenvoices for speaker adaptation," in *Fifth International Conference on Spoken Language Processing,* 1998, .

[3] P. Kenny, M. Mihoubi and P. Dumouchel, "New MAP estimators for speaker recognition," in *Eighth European Conference on Speech Communication and Technology,* 2003, pp. 2961-2964.

[4] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2003, pp. 53-56.

[5] R. J. Vogt, B. J. Baker and S. Sridharan, "Modelling session variability in text independent speaker verification," in *INTERSPEECH,* 2005, pp. 3117-3120.

[6] D. E. Sturim, W. M. Campbell, Z. N. Karam, D. A. Reynolds and F. S. Richardson, "The MIT lincoln laboratory 2008 speaker recognition system," in *INTERSPEECH,* 2009, pp. 2359-2362.

[7] J. F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. Evans, B. Fauve and J. Mason, "ALIZE/SpkDet: A state-of-the-art open source software for speaker recognition," in *Proceedings of Odyssey,* 2008, .

[8] J. J. Remus, K. D. Morton, P. A. Torrione, S. L. Tantum and L. M. Collins, "Comparison of a distance-based likelihood ratio test and k-nearest neighbor classification methods," in *IEEE Workshop on Machine Learning for Signal Processing,* 2008, pp. 362-367.

[9] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, pp. 27:1-27:7, 2011.

[10] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.

[11] W. M. Campbell, D. E. Sturim and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters,* vol. 13, pp. 308-311, 2006.

[12] D. V. Nguyen and D. M. Rocke, "Tumor classification by partial least squares using microarray gene expression data," *Bioinformatics,* vol. 18, pp. 39-50, January 1, 2002, 2002.

[13] J. Gottfries, K. Blennow, A. Wallin and C. G. Gottfries, "Diagnosis of Dementias Using Partial Least Squares Discriminant Analysis," *Dementia and Geriatric Cognitive Disorders,* vol. 6, pp. 83, 1995.

[14] J. B. Tenenbaum, V. d. Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, pp. 2319-2323, 2000.

[15] S. de Jong, "SIMPLS: An alternative approach to partial least squares regression," *Chemometrics Intellig. Lab. Syst.,* vol. 18, pp. 251-263, 1993.

[16] B. V. Srinivasan, D. Garcia-Romero, D. N. Zotkin and R. Duraiswami, "Kernel partial least squares for speaker recognition," in *INTERSPEECH,* 2011, pp. in press.

APPENDIX

First NormFeat call configuration parameters:

```
mode norm
bigEndian false
loadFeatureFileFormat SPRO4
saveFeatureFileFormat SPRO4
loadFeatureFileExtension .prm
saveFeatureFileExtension .norm.prm
featureServerBufferSize ALL_FEATURES
sampleRate 8000
labelSelectedFrames speech
segmentalMode false
writeAllFeatures true
frameLength 0.02
vectSize 32
featureServerMode FEATURE_WRITABLE
featureServerMemAlloc 500000000
addDefaultLabel true
defaultLabel speech
featureFilesPath ./feats/
verbose false
debug false
```

Energy Detector configuration parameters:

```
verbose false
verboseLevel 0
debug false
loadFeatureFileExtension .norm.prm
saveLabelFileExtension .lbl
loadFeatureFileFormat SPRO4
saveFeatureFileFormat SPRO4
saveFeatureFileSPro3DataKind FBCEPSTRA
minLLK -200
maxLLK 200
bigEndian false
featureServerBufferSize ALL_FEATURES
labelOutputFrames speech
frameLength 0.02
% featureServerMask 0-31
vectSize 32
labelSelectedFrames all
addDefaultLabel true
defaultLabel all
nbTrainIt 8
segmentalMode file
varianceFlooring 0.0001
varianceCeiling 1.5
mixtureDistribCount 10
baggedFrameProbabilityInit 0.001
thresholdMode weight
alpha 0.00
```

Second NormFeat call configuration parameters:

```
mode norm
bigEndian false
loadFeatureFileFormat SPRO4
saveFeatureFileFormat SPRO4
loadFeatureFileExtension .norm.prm
saveFeatureFileExtension .mfcc
featureServerBufferSize ALL_FEATURES
sampleRate 8000
labelSelectedFrames speech
addDefaultLabel true
defaultLabel speech
segmentalMode false
frameLength 0.02
vectSize 32
featureServerMode FEATURE_WRITABLE
featureServerMemAlloc 500000000
writeAllFeatures false
loadFeatureFileVectSize 32
saveLabelFileExtension .lbl
labelFilesPath ./labels/
verbose false
debug false
```

TrainTarget configuration parameters:

```
inputWorldFilename TRAINED_WORLD
gender M
bigEndian false
featureServerMemAlloc 10000000
featureServerBufferSize ALL_FEATURES
featureServerMode FEATURE_WRITABLE
frameLength 0.02
sampleRate 8000
writeAllFeatures true
segmentalMode false
debug false
saveMixtureFileFormat RAW
loadMixtureFileFormat RAW
loadMixtureFileExtension .gmm
saveMixtureFileExtension .gmm
loadFeatureFileFormat SPRO4
loadFeatureFileExtension .mfcc
loadMatrixFormat DB
saveMatrixFormat  DB
%loadMatrixFilesExtension .matx
%saveMatrixFilesExtension .matx
%vectorFilesextension .sv
%featureServerMask 0-18,20-50
%loadFeatureFile
addDefaultLabel true
defaultLabel speech
labelSelectedFrames speech
normalizeModel false
mixtureFilesPath ./gmm/
%matrixFilesPath ./mat/
%vectorFilesPath ./svec/
%featureFilesPath E:\data\Abacus_MFCC\2006\train\
computeLLKWithTopDistribs COMPLETE
topDistribsCount 10
maxLLK 200
minLLK -200
nbTrainIt 1
MAPAlgo MAPOccDep
meanAdapt true
MAPRegFactorMean 14.0
regulationFactor 14.0
%targetIdList ./ndx/target_male_1conv4w.2006.ndx
channelCompensation none
saveMixture true
```

TrainWorld configuration parameters:

```
featureFilesPath ./
mixtureFilesPath ./gmm/
labelFilesPath ./labels/
loadMixtureFileFormat RAW
loadMixtureFileExtension .gmm
saveMixtureFileFormat RAW
saveMixtureFileExtension .gmm
loadFeatureFileFormat SPRO4
loadFeatureFileExtension .norm.prm
bigEdian false
featureServerBufferSize ALL_FEATURES
distribType GD
frameLength 0.02
vectSize 32
labelSelectedFrames speech
debug true
verbose true
fileInit false
%inputWorldFilename xxx
normalizeModel true
mixtureDistribCount 500
baggedFrameProbabilityInit 0.08
maxLLK 200
minLLK -200
baggedFrameProbability 0.1
nbTrainIt 6
initVarianceFlooring 0.5
initVarianceCeiling 10.0
nbTrainFinalIt 4
finalVarianceFlooring 0.5
finalVarianceCeiling 10.0
numThread 2
verbose false
debug false
```

ComputeTest configuration parameters:

```
bigEndian                            false
featureServerMemAlloc                10000000
featureServerBufferSize              ALL_FEATURES
featureServerMode                FEATURE_WRITABLE
frameLength                          0.02
sampleRate                           8000
writeAllFeatures                     true
segmentalMode                        false
debug                                true


*****************************************************************************
*   In & Out
*****************************************************************************
saveMixtureFileFormat                RAW
loadMixtureFileFormat                RAW
loadMixtureFileExtension             .gmm
saveMixtureFileExtension             .gmm

loadFeatureFileFormat                SPRO4
loadFeatureFileExtension             .mfcc

loadMatrixFormat                     DB
saveMatrixFormat                     DB

loadMatrixFilesExtension             .matx
saveMatrixFilesExtension             .matx

vectorFilesExtension                 .sv


*****************************************************************************
*   Path
*****************************************************************************
mixtureFilesPath                     ./gmm/
matrixFilesPath                      ./mat/
vectorFilesPath                      ./svec/
% featureFilesPath
C:\Users\Jenniffer\Desktop\MistralWin32\MistralWin32\Lia_Spk_Det\


********************************************************
*      Feature options
********************************************************
% featureServerMask                  0-18,20-50
vectSize                             32
loadFeatureFileBigEndian             false
addDefaultLabel                      true
defaultLabel                         speech
labelSelectedFrames                  speech
normalizeModel                       false
ndxFilename ndx.lst
```

```
*****************************************************************************
*      Computation
*****************************************************************************
computeLLKWithTopDistribs               COMPLETE
topDistribsCount                    10
maxLLK                                  200
minLLK                                  -200
nbTrainIt                       1

labelSelectedFrames                     speech
normalizeModel                      false

MAPAlgo                         MAPOccDep
meanAdapt                       true
MAPRegFactorMean                    14.0
regulationFactor                    14.0

*****************************************************************************
*      ComputeTest Specific Options
*****************************************************************************
% ndxFilename                       .\ndx
outputFilename                      score.res


% channelCompensation                   none
 inputWorldFilename                     TRAINED_WORLD
gender M
featureFilesPath ./feats/
```

Main MATLAB script for generating cross-condition EER matrices

```
UBM_Training='/home/speakerid/MultiRoom8/Development';

folders={'Condtion4_Enroll-Sm4/train','Condtion7_Med5-Sm5/test',...
    'Condtion7_Med5-Sm5/train','Condtion1_Lg5-Sm4/train',...
    'Condtion3_Enroll-Sm6/test','Condtion8_Med2-MultiTrans/train',...
    'Condtion5_Med3-Sm3/test','Condtion5_Med3-Sm3/train',...
    'Condtion2_Sm4-Lg5/train','Condtion6_Lg4-Med5/train'};

SPRO='/home/speakerid/spro-4.0/sfbcep';
SPROTxt='/home/speakerid/spro-4.0/scopy';

% NormFeat Directory Paths:
NormFeatExe='/usr/local/LIA_RAL/2.0/bin/NormFeat';
NormFeatConfig='/home/ALIZEToolbox/NormFeat.cfg';
NormFeatConfig2='/home/ALIZEToolbox/NormFeat_energy.cfg';

% EnergyDetector Directory Paths:
EnergyDetectExe='/usr/local/LIA_RAL/2.0/bin/EnergyDetector';
EnergyDetectConfig='/home/ALIZEToolbox/EnergyDetector.cfg';

% UBM and GMM Training Directory Paths:
UBMExe='/usr/local/LIA_RAL/2.0/bin/TrainWorld';
UBMConfig='/home/ALIZEToolbox/TrainWorld.cfg';

GMMExe='/usr/local/LIA_RAL/2.0/bin/TrainTarget';
GMMConfig='/home/ALIZEToolbox/TrainTarget.cfg';

SVExe = '/usr/local/LIA_RAL/2.0/bin/modelToSv';
SVConfig = '/home/ALIZEToolbox/modelToSv.cfg';

computeTestExe='/usr/local/LIA_RAL/2.0/bin/ComputeTest';
computeTestConfig='/home/ALIZEToolbox/ComputeTest.cfg';

EER=nan(length(folders));

TrainUBM(UBM_Training,SPRO,SPROTxt, NormFeatExe,...
    NormFeatConfig, NormFeatConfig2, EnergyDetectExe,...
    EnergyDetectConfig, UBMExe, UBMConfig)

 matlabpool open

% extract all the features, learn all the GMMs
parfor i = 1:length(folders)
    GMMprocess=['/home/speakerid/MultiRoom8/' folders{i} '/seg1'];

TrainGMM(UBM_Training,GMMprocess,...
    SPRO, SPROTxt, NormFeatExe, NormFeatConfig,...
    NormFeatConfig2, EnergyDetectExe, EnergyDetectConfig,...
    GMMExe, GMMConfig)

% extract supervectors
SupVect(GMMprocess,SVExe,SVConfig);
```

```matlab
    GMMprocess=['/home/speakerid/MultiRoom8/' folders{i} '/seg2'];

TrainGMM(UBM_Training,GMMprocess,...
    SPRO, SPROTxt, NormFeatExe, NormFeatConfig,...
    NormFeatConfig2, EnergyDetectExe, EnergyDetectConfig,...
    GMMExe, GMMConfig)

% extract supervectors
SupVect(GMMprocess,SVExe,SVConfig);

end


for traincondition= 1:length(folders)
parfor testcondition = 1:length(folders)
    GMM_Test=['/home/speakerid/MultiRoom8/' ...
        folders{testcondition} '/seg2'];
    GMM_Train=[' /home/speakerid/MultiRoom8/' ...
        folders{traincondition} '/seg1' ];

FID = int2str(100*traincondition + testcondition);

[FID] = ComputeDecisionMetrics(computeTestExe, computeTestConfig,...
    GMM_Test,GMM_Train, FID);

[EER(traincondition,testcondition)]=Scoring(computeTestExe, ...
    computeTestConfig, GMM_Test,...
    GMM_Train, 1,FID);


end
end
```

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

CDDR: Classification Directed Dimensionality Reduction
DET: Detection Error Trade-off
EER: Equal-Error Rates
EMAP: Extended Maximum A Posteriori
GMM: Gaussian Mixture Model
GMM-UBM: Gaussian Mixture Model – Universal Background Model
GMMSV-NN: GMM supervector features with nearest neighbor classifier
GMMSV-RF: GMM supervector features with Random Forest classifier
GMMSV-SVM: GMM supervector features with linear-kernel Support Vector Machine
LDA: Linear Discriminant Analysis
LFA: Latent Factor Analysis
MAP: Maximum A Posteriori
MFCC: Mel-Frequency Cepstral Coefficients
NIST: National Institute of Standards and Technology
NN: Nearest Neighbor
PCA: Principal Component Analysis
PLS: Partial Least Squares
PLS-NN: Partial Least Squared decomposed supervector with nearest neighbor classifier
PLS-RF: Partial Least Squared decomposed supervector with Random Forest classifier
PLS-SVM: Partial Least Squared decomposed supervector with radial-basis kernel Support Vector Machine
PLSDA: Partial Least Squares Discriminant Analysis
RBF: Radial Basis Function
RF: Random Forest
SVM: Support Vector Machine
UBM: Universal Background Model